Flow-based Enterprise Process Modeling

Sabah Al-Fedaghi

Kuwait University
sabah@alfedaghi.com

Abstract

Business process modeling is an approach that builds visual representations of business activities, events, flow controls, functions, applications, stakeholders, and their relationships. It has been shown that such modeling significantly benefits an enterprise by providing an instrument for documentation, communication, and management purposes. Extensive work has been published in the area of enterprise modeling in various fields; nevertheless, deficiencies still exist in tools and methods for capturing complex business processes. In this context, the Architecture of Integrated Information Systems (ARIS) is a system for development and modeling that has been used as a first step in developing a business process model. This paper examines ARIS by contrasting it with a newly proposed flow-based diagrammatic method called the Flowthing Model (FM). Results show that FM is a viable tool for application to business process modeling.

Keywords: *conceptual model, Architecture of Integrated Information Systems, Enterprise Modeling, requirements specification*

1. Introduction

According to the publishers of GERAM [1],

One of the most important characteristics of today's enterprises is that they are facing a rapidly changing environment and can no longer make predictable long term provisions. This necessitates the integration of the enterprise operation and the development of a discipline that organizes all knowledge that is needed to identify the need for change in enterprises and to carry out that change expediently and professionally. This discipline is called Enterprise Engineering.

Reference architecture, in this context, is a generalized framework for a structure that describes components and utilizes *enterprise engineering* to model an enterprise [2–6]. Enterprise modeling focuses on flows of information, control, and materials among organizational units within a single enterprise to improve interoperability. It involves development of an abstract representation of the organization's structure, analysis of its business processes, and building of an information system infrastructure. With this perspective, *process modeling* is an approach to graphically representing business activities, events, flow controls, stakeholders, and their relationships. It is an important instrument for documenting business operations and facilitating communication between relevant stakeholders, and for automation as well as execution purposes. Additionally, business process modeling enables process change decisions to address cost, compliance, and efficiency issues [7]. Studies have shown that process modeling is important for effective communication within an enterprise [5, 8]. It has also been demonstrated that visualizing business processes through modeling results in significant benefits [8].

Since the nineties, extensive work has been published in the area of process modeling in the field of enterprise engineering (*e.g.*, [9–13]); nevertheless, “the lack of powerful tools as well as methodological deficiencies – particularly with regard to capturing complex process logics and dynamics – are major obstacles for a successful re-engineering of business processes” [14].

This paper introduces a general diagrammatic tool for modeling processes and, without loss of generality, contrasts it with the diagrammatic tools used in the Architecture of Integrated Information Systems (ARIS). This focus of comparison provides an initial appraisal of our new method with reference to a well-known process modeling methodology.

ARIS is the architecture for development and modeling in this area [15]. It can be used as “a keystone for Business Process Reengineering” [16]. Also, it is “the first step to develop a business process model containing all basic features for describing business processes“ [17]. “The ARIS-approach not only provides a generic and well documented methodological framework but also a powerful business process modeling tool. This tool supports the entire process re-engineering project during all life cycle phases” [14].

ARIS focuses on the requirements phases, using process chain diagrams, in the design of information systems, with a multiview approach in which each view is further detailed according to the stages of the software life cycle [14][18]. Views include functional view (transforming input into output), data view (processing and triggering), process view, and organizational view.

An example of an ARIS description of views that involves processes is shown in Figure 1 [17].

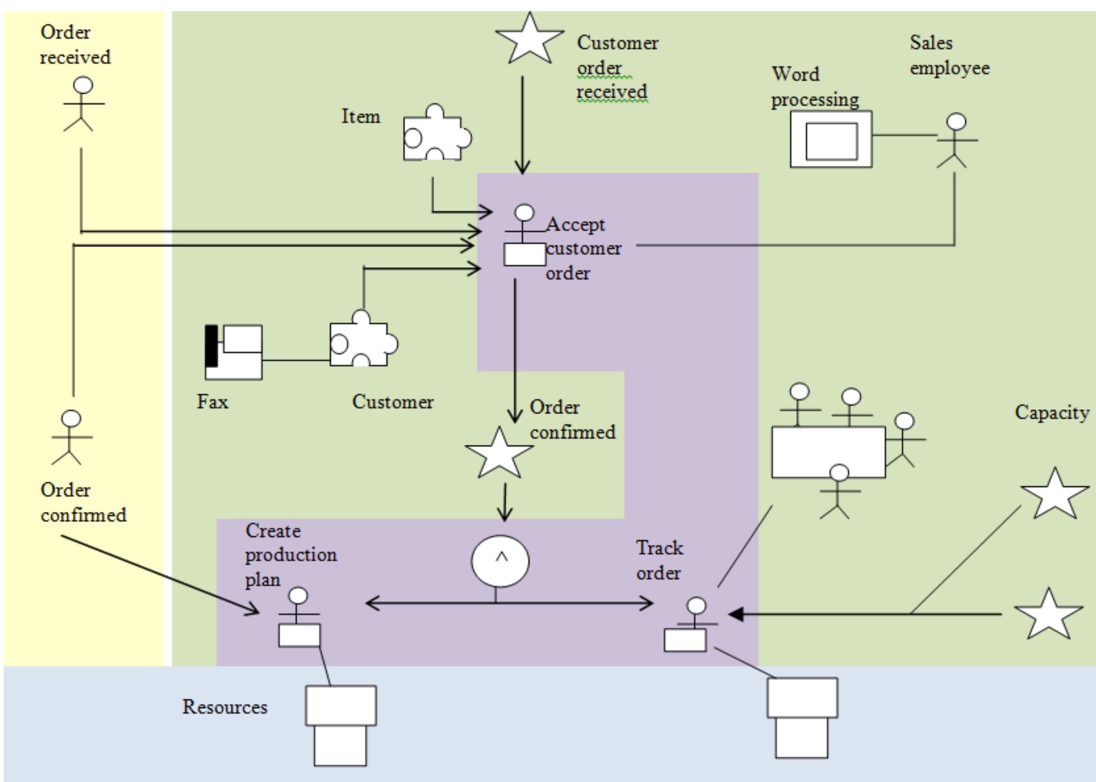


Figure 1. Process Model Views (from [17])

A **Customer order** is a received event that activates the **Accept customer order** procedure. The current state of the relevant process environment is first described in terms of data relating to customers and items. The procedures are executed by sales employees. Upon completion of the **Accept customer order** procedure, a **confirmation** event occurs that triggers other procedures. The **Accept customer order** function generates a product that is used as an input in processing of subsequent procedures [17]. Occurrence of an Event (*e.g.*, Customer order received or Invoice created) indicates that the state of data has changed.

To reduce complexity, the diagram is broken down into individual views by use of different colors. The views are depicted such that only a few relationships are shown between them, but as a result of such separation into individual views, "the process component relationships across the views are lost. For this reason, the Control view is provided as an additional view for describing the relationships between views" [17]. Also, ARIS utilizes process chain diagrams to describe an overview of the information system under consideration, as shown in Figure 2 [19].

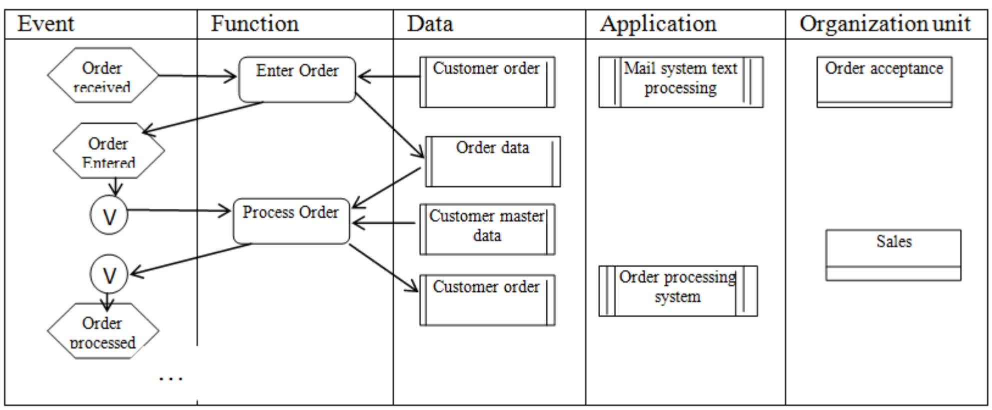


Figure 2. Partial View of a Process Chain Diagram (partial, from [17])

While such a brief description and diagrams of ARIS do not give a complete picture of the approach, it is nevertheless sufficient for our purpose in this paper: *We contrast ARIS requirements specification methodology with our flow-based diagrammatic approach, called the Flowthing Model (FM), in order to demonstrate that the FM method provides a promising systematic conceptual picture of enterprise processes.* Experiments with FM through examples from ARIS may lead to the integration of FM into mainstream research in software engineering (*e.g.*, object oriented, UML [20]), and they may also influence its direction.

Accordingly, this paper demonstrates that the proposed FM diagrammatic method can capture the interweaving of different events using the notion of *flow*. Flow here refers to the flow of things, as in the specification of flows of electricity, water, gas, signals (*e.g.*, telephone lines), added to the blueprint of a high-rise building. ARIS's Figure 1 will be discussed when we redraw the example in terms of the flow-based FM method.

While we focus on ARIS in this paper, our FM can be applied to most current diagramming methodologies (*e.g.*, UML) as a tool for high-level specifications of general information systems, as demonstrated in several previous publications (*e.g.*, [21–23]). For the sake of a self-contained paper, the next section briefly reviews FM, while the example using horizontal swimlanes to model business processes, also shown in the next section, is a new contribution.

2. Flowthing Model

The Flowthing Model (FM) is a uniform method for representing "things that flow," called *flowthings*. Flow in FM refers to the exclusive (*i.e.*, being in one and only one) transformation among six *states* (also called stages): transfer, process, creation, release, arrival, and acceptance, as shown in Figure 3. We will use *Receive* as a combined stage of *Arrive* and *Accept* whenever arriving flowthings are always accepted.

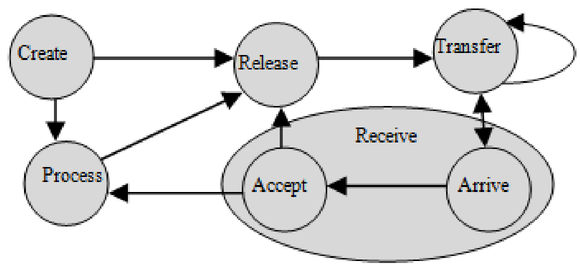


Figure 3. Flowsystem

The fundamental elements of FM are as follows:

Flowthing: A thing that has the capability of being created, released, transferred, arrived, accepted, and processed while flowing within and between "units" that are called *spheres*.

A flow system (referred to as *flowsystem*) is a system with six stages and transformations (edges) between them (Figure 2).

Spheres and subspheres: These are the environments of the flowthing, such as, *e.g.*, a company, a computer, and a person.

Triggering: Triggering is a transformation (denoted by a dashed arrow) from one flow to another, *e.g.*, a flow of electricity triggers a flow of water.

In addition to the fundamental characteristics of flow in FM, the following types of possible operations exist in different stages:

1. *Copying*: Copy is an operation such that *flowthing* f => f. That is, it is possible to copy f to produce another flowthing f in a system S. In this case, S is said to be S *with copying* feature, or, for short, *Copy S*. For example, any informational flowsystem can be copy S, while physical flowsystems are non-copying S. Notice that in copy S, stored f may have its copy in a non-stored state. It is possible that copying is allowed in certain stages and not in others.

2. *Erasure*: Erasure is an operation such that *flowthing* f => e, where e denotes the *empty flowthing*. That is, it is possible to erase a flowthing in S. In this case, S is said to be S *with erasure* feature, or, for short, *erasure S*. Erasure can be used for a single instance, all instances in a stage, or all instances in S.

3. *Canceling*: *Anti-flowthing* f^- (f with superscript –) is a flowthing such that (f^- + f) => e, where e denotes the empty flowthing, and + denotes the presence of f^- and f.

It is possible that the anti-flowthing f^- is declared in a stage or a flowsystem. If flowthing f triggers the flow of flowthing g, then anti-flowthing f^- triggers anti-flowthing g^-.

An example of the use of these FM features is erasure of a flow, as in the case of a customer who orders a product, then cancels the order, an action that might require the cancellation of several flows in different spheres triggered by the original order.

Formally, FM can be specified as $FM = \{S_i (\{F_j\}, T_l), \{(F_{ij}, F_{ij})\}, 1 \le i \le n, 1 \le j \le m, 1 \le l \le t\}$ That is, FM is a set of spheres $S_1, ...S_n$, each with its own flowsystems $F_{ij},... F_{im}$. T is a type of flowthing $T_1,..., T_t$. Also, F is a graph with vertices V that is a (possibly proper) subset {Arrive*, Accept*, Process*, Create*, Release*, Transfer*}, where the asterisks indicate secondary stages. For example, {Copy, Store, and Destroy} can represent these secondary stages.

Example: UML activity diagram usage includes exploring the logic of complex rules and processes. An example of this use given by [24] shows horizontal swimlanes representing business processes, as shown in Figure 4.

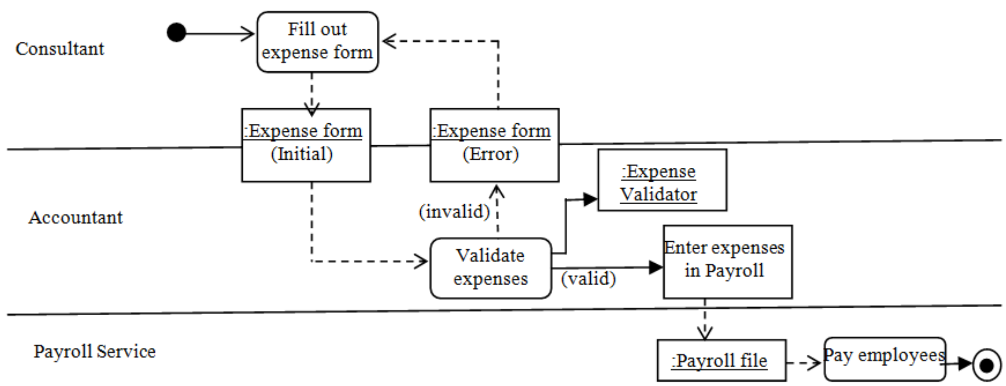


Figure 4. Swimlanes Depicting Submitting of Expenses (from [24])

Figure 5 shows the corresponding FM representation of this process. Three spheres are involved: Consultant, Accountant, and Payroll service. The Consultant's sphere comprises two flowsystems: form and payment. That of the Accountant has one flowsystem: form, and the sphere of Payroll service has one flowsystem: payroll. The process begins when the consultant creates (circle 1 in the figure) an expense form that flows (2) to the accountant. The accountant processes (3) the form; if an error is found (4), the form is returned to the consultant (5), where it is processed (6), triggering (7) the creation of another form. If the form is valid (8), it triggers (9) the creation (10) of an expense payment in the payroll system. The payment then flows to the consultant.

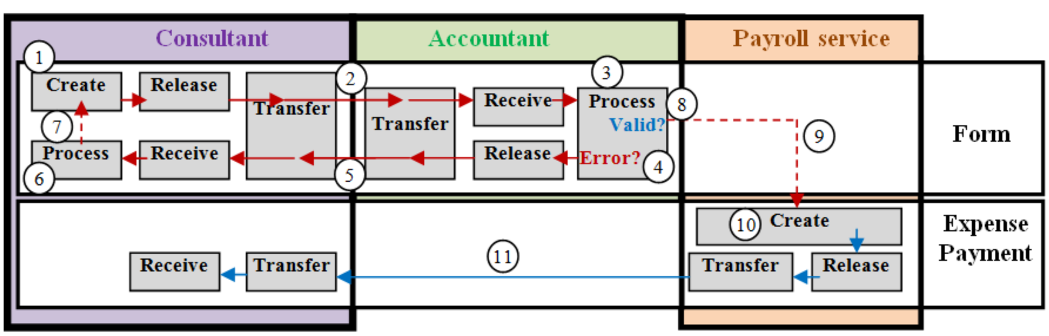


Figure 5. FM Representation of the Expense Example Shown in Figure 4

3. FM Enterprise Modeling

The ARIS methodology of modeling requirements, as illustrated in Figures 1 and 2, appears to lack a logical thread connecting different streams of events. That is, a precedence spine in the chain of the ordering process appears to be missing. Figure 1 is a mix of conceptual notions (capacity, resources) and entities (persons, items) with technical terms added (*e.g.*, word processing, fax). The arrows confuse the conceptual picture, with connections among customer, item, procedures, and capacity all represented by the same type of arrow—which can indicate flow of control, flow of things, and other flows and events. This becomes clear when Figure 1, as we interpret it, and also Figure 2 are compared with the FM representation shown in Figure 6.

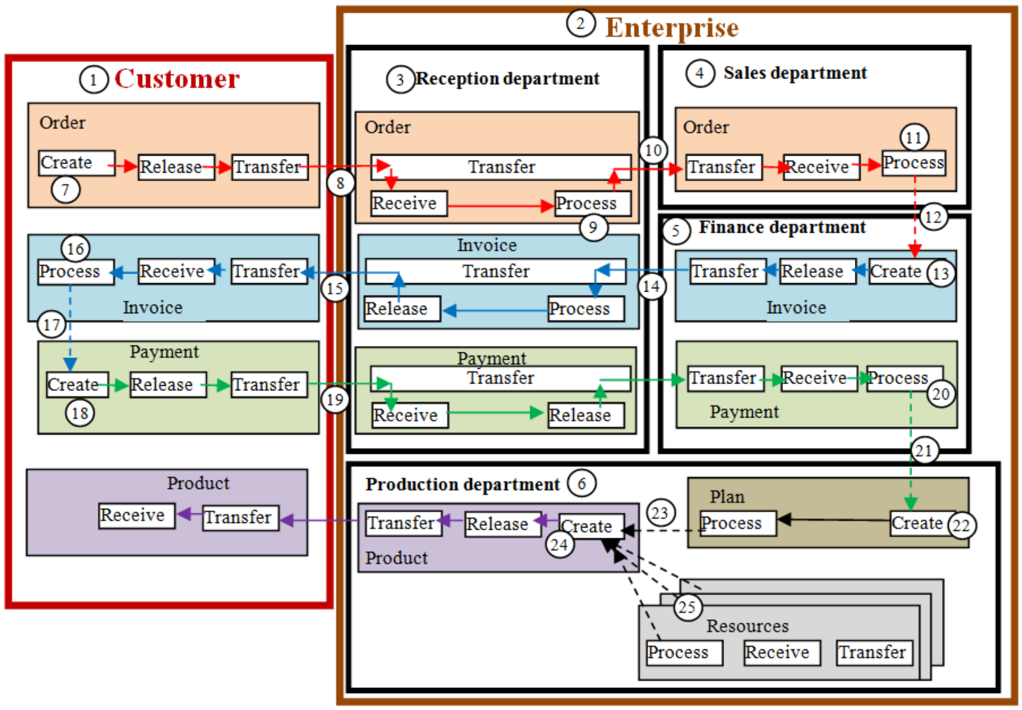


Figure 6. FM Representation Corresponding to Figure 1

Figure 6 comprises two spheres: Customer and Enterprise (circles 1 and 2, respectively). The figure has been drawn according to our understanding of Figure 1 and includes breaking the enterprise into four subspheres: Reception, Sales, Finance, and Production departments. The Reception department handles any type of customer communication, which is represented by separating the individual who "accepts customer order" (middle of Fig. 1) from the Sales employee in the same figure. Also, we interpret "Order confirmed" in Figure 1 to mean sending invoices and confirming payments.

In the FM representation (Figure 6), we can recognize six streams of flow that trigger each other, as illustrated in Figure 7. This is a very coherent conceptual depiction when compared with the spread of happenings shown in Figure 1.

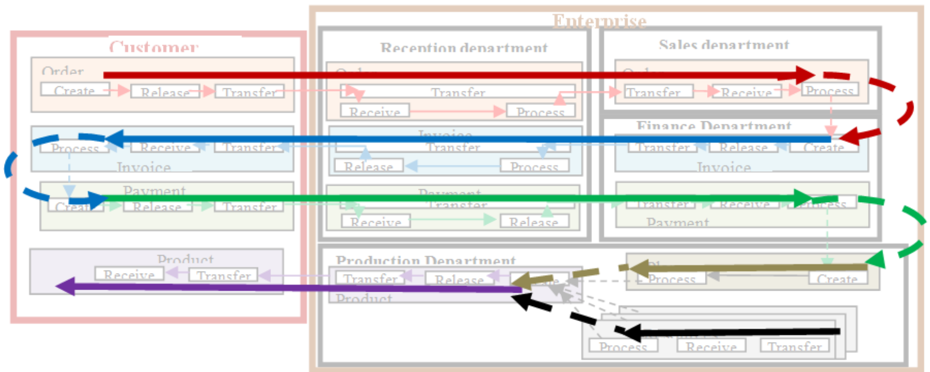


Figure 7. Streams of Flow in the FM Representation Shown in Figure 6

Thus, beginning at circle 7 in Figure 6, a customer creates an order that flows to the Reception department (8), where it is processed (9), This is then forwarded to the Sales department (10), where it is processed (11), triggering (12) the creation of an invoice (13) in the Finance department that flows to the Reception department (14), which in turn sends it to the Customer (15). The customer processes the invoice (16), and that triggers (17) the creation of a payment (18) that flows to the Enterprise (19), to be processed in the Finance department (20). Assuming the payment is acceptable, this triggers (21) the creation and processing of a production plan (22) which then triggers creation of the ordered product (23). We assume that this creation of a product "pulls up" resources and at the end triggers (24) completion of the product. The finished product then flows to the customer.

It is possible to complete some of the details of this conceptual picture by utilizing the same methodology. For example, completing the flow of the order in the case of rejection during processing to trigger sending a rejection note, or sending a copy of the order to the Production department. In the next section, we will add a flow of product lists from the Enerprise to the customer, and a flow of information about customers to the Sales department.

4. View Mapping

In the ARIS methodology, a process chain diagram (PCD), as shown in Figure 2, "represents a closed process chain. All views of a business process (organization view, data view, function view, resource view) are expressed with their relationships in a coherent form" [17]. Events in PCD include "Order received", "Order entered", "Order processed", and "order updated". Instead of another diagram that tries to interweave events, functions, data, applications, and organizational units, we find in Figure 6 a continuous chronology:

An Order is created, released, and transferred to the Reception department, where it is received and processed, to be sent to the Sales department, where it is received and processed, and if it is accepted in processing, the Sales department is triggered to …

We can add a conceptual description of the interior of different stages. For example, *process* in the Sales department might include, "if order is accepted, then trigger creation of an invoice, else trigger something else".

For example, suppose the Customer needs the Enterprise to provide a list of products. It is possible to incorporate this need, as shown in Figure 8. In this case, the customer's screen of orders (shown in Figure 9) is constructed from two flows, the order flow as discussed in regard to Figure 6, and the flow of product listings from, say, the database of the Enterprise, shown in Figure 8. Thus, Figure 6 can be extended to incorporate this last flow. For the sake of brevity, only the upper part of the extended figure is shown in Figure 8. To construct the screen, the list of products is retrieved from the database (circle 1 in Figure 8), and flows to the customer (2), to be processed in construction of the screen (3).

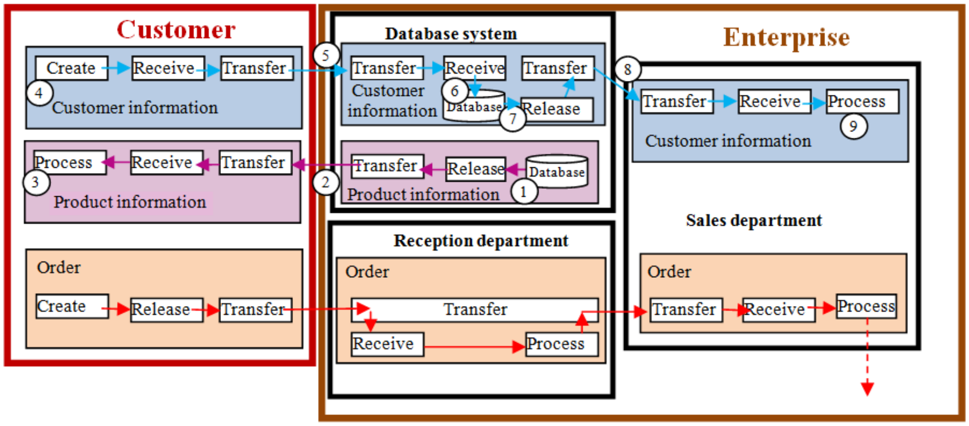


Figure 8. Extending the Upper Part of the FM Representation of Figure 6

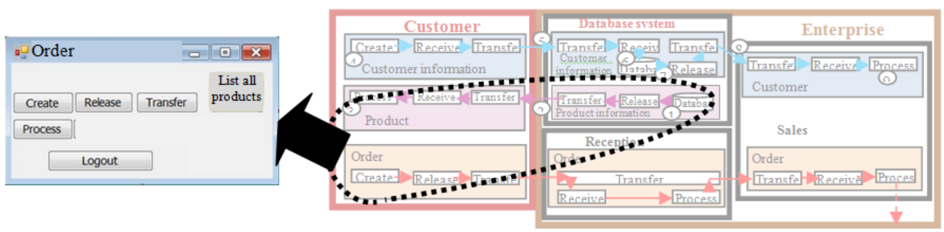
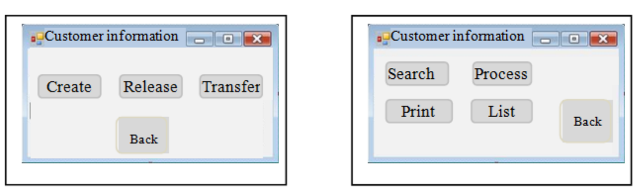


Figure 9. Customer's Screen for Orders

Now suppose the Enterprise's database system requires entering data about the Customer for use by the Sales department in processing orders. In this case, a customer information flow is added, as shown in Figure 8. Thus, the customer inputs (creates; circle 4 in Figure 8) his/her information through a screen such as the one shown in Figure 10(a). The information flows to the database system (5), where it is stored in the database (6). The Sales department can view a screen such as the one shown in Figure 10(b), constructed from information retrieved from the customer (7) that flows to the Sales department (8), where it is processed (9).



(a) Customer's screen for Orders (b) Sales department's screen for Orders

Figure 10. Extending the Upper part of the FM Representation of Figure 6 to Incorporate Entering of Customer Information

Note how FM is applied uniformly across the system development cycle from the specification of requirements up to and including design of user interfaces (screens). What remains is the issue of implementing the different stages within flowsystems. For example, for Figure 10(b), where a user in the Sales department accesses Customer information, the implementation issue involves “activating” process in the screen, assuming that one particular customer has already selected. This results in:

- A request from the SalesManager.CustomerInformation module (flowsystem *customer information* in the Sales department, Figure 8) to the DataBase.CustomerInformation module (flowsystem: *Customer information* in Database system in Figure 8) to retrieve that particular customer’s information.
- The SalesManager.CustomerInformation module retrieves the required data, then releases and transfers it to the SalesManager.CustomerInformation module.
- Arrival of data at the SalesManager.CustomerInformation module results in a display of customer information for the user in the Sales department.

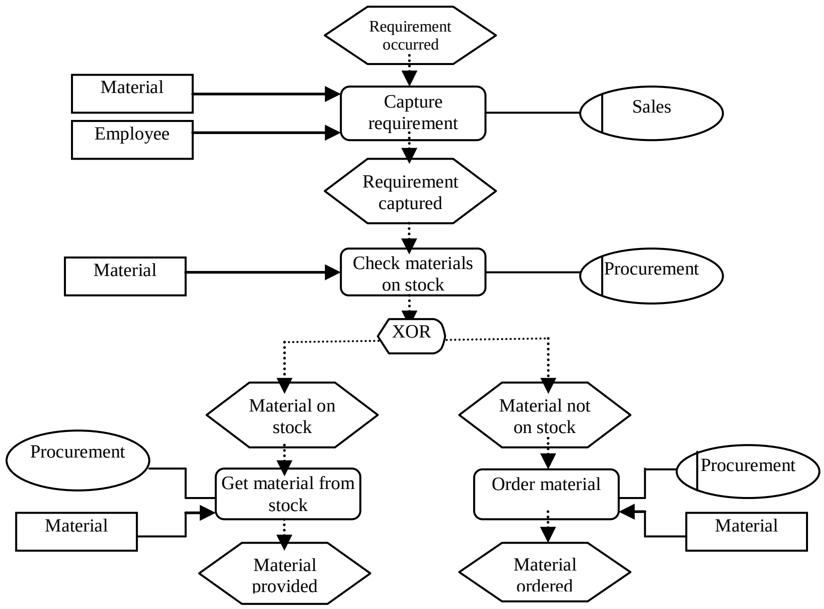
5. Contrasting Diagrams

Within the framework of ARIS, another diagram called the Event-driven Process Chain (EPC) has been embedded [25]. Generally, EPC and UML diagrams have been developed in the frameworks of business process modeling and software development, respectively [26].

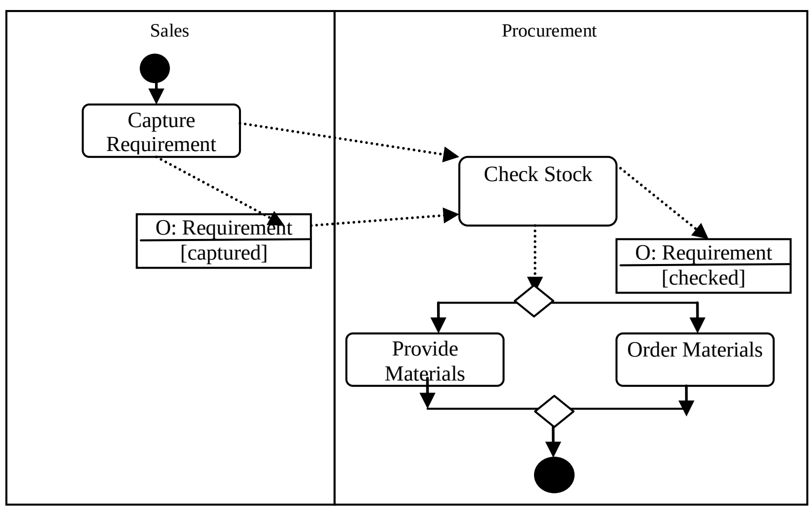
The strength of EPC lies on its easy-to-understand notation that is capable of portraying business information system while at the same time incorporating other important features such as functions, data, organizational structure and information resources as already described before. This makes EPC a widely acceptable standard to denote business processes. [26]

Several works have been published comparing these two diagrammatic methodologies (*e.g.*, [26–28]). Comparisons may involve syntax, semantics, and other aspects. Figure 11(a) depicts a sample EPC diagram showing an excerpt from procurement logistics, and Figure 11(b) shows its corresponding UML activity diagram. These diagrams are contrasted in terms of visualization of processes and workflows, functions, events, and control flow. In 2001, it was argued that an integrated approach is desirable:

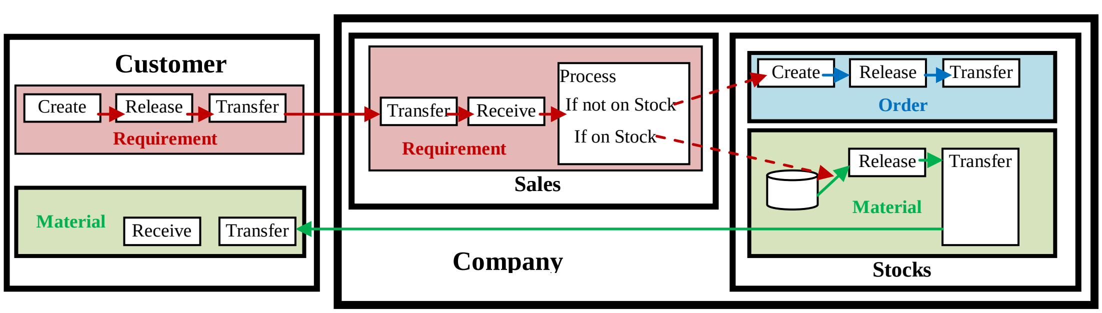
Integrating the strengths and advantages of the EPC and the UML create[s] a method that covers both areas of business process modeling and object-oriented information system. The purpose of the integration approach is on the one hand to preserve the potential and the end user’s acceptance of the standard EPC method for business process modeling and on the other hand to integrate the object-oriented methods with the EPC [26].



(a) Example of an EPC diagram showing an excerpt from procurement logistics (from [26])



(b) Example of an activity diagram showing an excerpt from procurement logistics (from [26])



(c) FM representation

Figure 11. EPD, Activity and FM Diagrams

In 2006, Storrle [28] concluded that “while earlier versions of the UML did not provide sufficient means for modeling of business processes, the current version does.” Since UML provides additional benefits over EPCs in software development, Storrle [28] predicted that for business process modeling, activity diagrams would prevail over EPCs. In 2009, Gross *et al.*, [27] set up experiments examining both activity diagrams and EPCs from the perspective of requirements engineering. The results show that activity diagrams perform better than EPCs. A report in 2011 about UML activity diagrams used in the modeling of business processes indicated achieving “a significantly better comprehension level” in modeling of business processes when a “lighter” variant of the activity diagrams was used, with “no significant impact on the effort to accomplish the tasks” [29]. FM has already been contrasted with the UML activity diagram in several publications [21–23]. It is interesting to compare the three diagrammatic methodologies—EPC, activity, and FM diagrams—by illustrating the same problem with them. Accordingly, Figure 11(c) shows the FM representation of the problem shown in Figure 11(a) and (b). FM seems viable as a model for application to business process modeling.

6. Conclusion

This paper introduces a flow-based diagrammatic methodology called the Flowthing Model (FM) for use as a tool in business process modeling. In exploring this idea, and without loss of generality, the paper concentrates on contrasting FM with the diagrammatic tools used in the Architecture of Integrated Information Systems (ARIS). Such a focus of comparison provides an initial appraisal of the new method with reference to a well-known process-modeling methodology. The results seem to indicate that the FM conceptual representation provides an alternative to other types of requirements specification depictions. To demonstrate its benefits, we developed examples side by side to depict the differences between three different diagrammatic methods. Of course, it is not possible to fully compare such a well-developed methodology as ARIS, with its long-proven history of application, to FM; nevertheless, this ought not to discourage exploring new approaches in the area of process modeling.

References

[1] GERAM, Generalised Enterprise Reference Architecture and Methodology Version 1.6.3. IFIP/IFAC Task Force on Architectures for Enterprise Integration, **(1999)** March. http://www.ict.griffith.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html

[2] P. Bernus and L. Nemes, “Modelling and Methodologies for Enterprise Integration”, Chapman & Hall **(1996)**.

[3] Lillehagen and Krogstie, State of the Art of Enterprise Modelling, Springer-Verlag, Berlin, Heidelberg **(2008)**, pp. 1-25.

[4] W. Bandara, G. G. Gable and M. Rosemann, “Factors and measures of business process modelling: model building through a multiple case study”, EJIS, vol. 14, no. 4, **(2005)**, pp. 347-360.

[5] M. Indulska, P. F. Green, J. Recker and M. Rosemann, “Business Process Modeling: Perceived Benefits”, ER 2009, Lecture Notes in Computer Science, Springer, vol. 5829, **(2009)**, pp. 458–471.

[6] J. Mendling, J. Recker and H. A. Reijers, “On the Usage of Labels and Icons in Business Process Modeling”, Int. J. Inform. Syst. Model. Des., vol. 1, no. 2, **(2010)**, pp. 40-58.

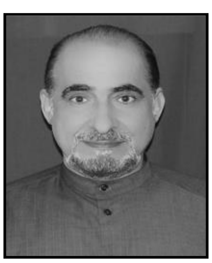
[7] N. H. Alzubidi, J. Recker and E. Bernhard, “A Study of the Use of Business Process Modelling at Suncorp”, Project Report No. 1, Queensland University of Technology, Brisbane, Australia, **(2011)** November.

[8] M. Kesari, S. Chang and P. Seddon, “A content analytic study of the advantages and disadvantages of process modeling”, The 14th Australasian Conference on Information Systems, Perth, Western Australia, **(2003)** November 26-28.

[9] C. J. Petrie, “Jr. Enterprise Integration Modelling”, ICEIMT Conference Proceedings, MIT Press, Cambridge, MA, **(1992)** (ISBN 0-262-66080-6).

[10] CIMOSA - Open System Architecture for CIM. ESPRIT Consortium AMICE, Springer-Verlag, Berlin, **(1993)** (ISBN 3-540-56256-7), (ISBN 0-387-56256-7).
[11] T. J. Williams, P. Bernous, J. Brosvic, D. Chen and G. Doumeingts, "Architectures for integrating manufacturing activities and enterprises", Computers in Industry, vol. 24, no. 2-3, **(1994)**, pp. 11-39.
[12] Y. Lu, H. Cai and L. Jiang, "Construction of BPMN-based Business Process Model Base Modelling Techniques for the Revised Business Process Reengineering", IJIIP: Int. J. Intell. Inform. Process, vol. 1, no. 2, **(2010)**, pp. 32-38.
[13] L. J. Menzli, S. A. Ghannouchi and H. H. Ben Ghezala, "A Guidance Process for the Selection of Business Process Modelling Techniques for the Revised Business Process Reengineering", JCIT: J. Convergence Inform. Technol. vol. 2, no. 2, **(2007)**, pp. 84-93.
[14] T. Williams, "Workflow Management within the ARIS Framework", http://www.pera.net/Methodologies/ARIS/ARIS_Paper_by_Ted_Williams.html.
[15] A. -W. Scheer, "ARIS: Business Process Frameworks", Springer **(1999)**.
[16] A.-W. Scheer, "Architecture of Integrated Information Systems", Springer, Berlin **(1992)**.
[17] Software AG, ARIS Method, ARIS Platform, Version 7.2, Service Release 2, **(2011)** October.
[18] F. Vernadat, "CIMOSA: Enterprise Modeling and Enterprise Integration Using a Process Based Approach", Proceedings of the JSPE/IFIP TC5/WG5.3 Workshop on the Design of Information Infrastructure Systems for Manufacturing, DIISM'93, Tokyo, **(1993)**, pp. 65-84.
[19] A.-W Scheer, "EDV-orientierte Betriebswirtschaftslehre", [EDP-oriented business management], Springer; 1st edition, **(1990)**, pp. 39.
[20] Z. Shi, P. Ai and Yuyan Guo, "Method of Business Modelling based on UML Extension", AISS: Adv. Inform. Sci. Serv. Sci., vol. 4, no. 13, **(2012)**, pp. 387-396.
[21] S. Al-Fedaghi, "Reconceptualization of Class-based Representation in UML", Int J. Comp. Sci., vol. 9, no. 5, **(2012)**.
[22] S. Al-Fedaghi, R. Alloughani and M. Al Sanousi, "Modeling Base for Process Specification in Workflows", J. Softw. Eng. Appl., vol. 5, **(2012)**, pp. 560-567.
[23] S. Al-Fedaghi, "Scrutinizing UML Activity Diagrams", 17th International Conference on Information Systems Development (ISD2008), Paphos, Cyprus, **(2008)** August, pp. 25-27.
[24] S. W. Ambler, "Agile Modeling: UML 2 Activity Diagramming Guidelines", Accessed November **(2012)**. http://www.agilemodeling.com/style/activityDiagram.htm.
[25] W. M. P. van der Aalst, "Formalization and verification of event-driven process chains", Computing Science Reports 98/01, Edited R. C. Backhouse and J.C.M. Baeten, Eindhoven University of Technology, Eindhoven, **(1998)**.
[26] Ferdian, "A Comparison of Event-driven Process Chains and UML Activity Diagram for Denoting Business Processes", Project Work, Information and Communication Systems, **(2001)**.
[27] F. Di Cerbo, G. Dodero, G. Reggio, F. Ricca and G. Scanniello, "Precise vs. Ultra-Light Activity Diagrams - An Experimental Assessment in the Context of Business Process Modelling", Product-Focused Software Process Improvement, Lecture Notes in Computer Science, 6759, **(2011)**, pp. 291-305.
[28] H. Störrle, "A Comparison of (e)EPCs and UML 2 Activity Diagrams", EPK, **(2006)**, pp. 177-188.
[29] A, Gross and J. Doerr, "EPC vs. UML Activity Diagram: Two Experiments Examining Their Usefulness for Requirements Engineering", 17th IEEE International Requirements Engineering Conference, **(2009)**, pp. 47-56.

Author



Sabah Al-Fedaghi holds an MS and a PhD in computer science from the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois, and a BS in Engineering Science from Arizona State University, Tempe. He has published two books and more than 150 papers in journals and conferences on software engineering, database systems, information systems, computer/information privacy, security and assurance, information warfare, and conceptual modeling. He is an associate professor in the Computer Engineering Department, Kuwait University. He previously worked as a programmer at the Kuwait Oil Company, where he also headed the Electrical and Computer Engineering Department (1991–1994) and the Computer Engineering Department (2000–2007).