System Integration: Case Study of an Airport Information System

Sabah Al-Fedaghi^{a,*}, Ahmed Haider^b

^aComputer Engineering Department, Kuwait University, P.O. Box 5969, Safat, 13060, Kuwait

^bDirectorate General of Civil Aviation, Kuwait Airport, P.O. Box 17, Safat, 13001, Kuwait

*Corresponding Author: sabah.alfedaghi@ku.edu.kw

Abstract

Current research in the field of *system integration* focuses on platforms on which various development models are combined into one integrated model that satisfies the need of stakeholders for views that can be used to describe, communicate, and develop systems. Motivations for such an approach include multidisciplinary applications, complexity, dynamics and change, micro/macro levels of analysis, and the crossing of the natural and human worlds. Nevertheless, an underlying tool for expressing the unified totality of a system's processes and concepts is lacking. This paper proposes a new approach to creating a basic framework in this domain; without loss of generality, it focuses on the case study of an airport information system.

Keywords: system specification, conceptual description, SysML, UML information system

1. Introduction

Information and communication technology can improve and refine the performance of diverse processes in organizations. *Integration* has been a central theme of research and an ongoing challenge in many fields such as requirements and software engineering, systems engineering, and information systems. Integration aims to maximize performance and productivity⁽¹⁾, facilitate exchange of resources within a system⁽²⁾, and attain coordination and monitoring capabilities.

Traditionally, the term *integration* refers to data integration⁽³⁾, functional integration⁽⁴⁾, and process integration⁽⁵⁾. New information and communication technologies and application areas have extended such a view of integration, especially with the arrival of new forms of information handling and communication networks such

as mobile devices, social media, electronic interactions (e.g., e-commerce), and virtual environments.

In engineering, *integration* often refers to the process of assembling subsystems into a system⁽⁶⁾. It is used also to denote the unification of standards, processes, and models. For example, Unified Modeling Language (UML) and SysML are platforms on which various developmental models have been combined into an integrated single model. The platform seems to satisfy the need of stakeholders for views that can be used to describe and communicate and to develop a product, e.g., software⁽⁶⁾, but according to Egyed⁽⁶⁾, "These views, unfortunately, do not exist. Instead, we are confronted with a number of loosely coupled, sometimes quite independent views." Multiple views often result in "inconsistency ... particularly if these views represent, say, different stakeholder perspectives or alternative design solutions"⁽⁷⁾.

In this paper, *integration* is viewed as an apparatus to facilitate interconnections and communication of processes within a system through a common *conceptual representation*. The focus is on integration of descriptions to provide a base for technical design specifications at the physical, system, and application levels. It is important in this context to develop a single *blueprint* for all processes in an integrated and uniform way. This blueprint needs a comprehensive description language that specifies all types of processes, e.g., physical, human, information, and engineering. It would be the description of an *extended "mechatronic-like" system* involving many actors and a multipart network of control and communication activities⁽⁷⁾.

Problem: In large-scale systems, processes are typically large in scale and great in complexity, especially with emerging networked applications such as in grids, air traffic management, and integrated supply chains.

Development of such systems is becoming more and more multidisciplinary, with many stakeholders, including client, architect, engineers, designers, contractors, and consultants, all of whom are dependent on each other for information. The impacts of failures in the design of these systems are quite costly. Difficulties in communication as well as in integration of subsystems have arisen from variations in representations of notions used by the various disciplines. A key problem in this context is lack of a unifying framework for development and management. “Lack of consideration of engineering projects as systems-of-systems has led to methodological challenges in creation of integrated tools and techniques for better analysis and management of complex projects”⁽⁸⁾. A great deal of current research has focused on methodological approaches to functional specifications, flow descriptions, and system structure definitions⁽⁹⁾ such as the model-based process⁽¹⁰⁾, in which the system is specified at various levels of granularity. Nevertheless, an underlying tool for expressing the unified totality of a system’s processes and concepts is lacking.

Needs: Accordingly, a need has arisen for new ways to address this problem in a systematic manner, incorporating the following features:

- Integration is applied through a conceptual apparatus that can be used to integrate multi-domain activities and interactions, e.g., physical, informational, technical, ...
- Complexity is resolved through simple, uniform notations applied across macro- and micro-levels of detail.
- Common understanding is facilitated through use of one language that can be used for specification (e.g., in contracts) as well as communication

Chen and Stroup⁽¹¹⁾ articulate five motivations for developing a unifying integrated framework for systems, based on the existence of multidisciplinary applications, complexity, dynamics and change, micro/macro levels of analysis, and the crossing of the natural and human worlds.

Solution: This paper proposes an approach to creating a basic framework and, without loss of generality, focuses on requirements handling in *information system* development. The underlying thesis of the paper is that the integration of different descriptions of a system needs an underlying tool for expressing the unified totality of a system’s processes and concepts, analogous to techniques in carpet weaving, where the weaver utilizes a ground fabric at the base of the design to bind, piece, and sew the patterns of fabric.

Note here that in a modeling approach such as UML and SysML, process integration is specified in terms of pieces (diagrams) woven together on a base of English, as

shown in Fig. 1. The model introduced in this paper can serve as a ground fabric for any type of application, e.g., requirements specification, software development, security binding, contract conditions, shared understanding, documentation ..., as seen in Fig. 2.

Fig. 1. UML-based specifications resting on a weak base of ambiguous natural language.

Fig. 2. FM description furnishes a base for specification

The paper proposes a conceptual representation in the form of a depiction of flows suitable for constructing holistic schemata that can be used for different applications, with the focus here on integration of several subsystems in the information system of an enterprise. The representation is essentially based on the notion of flow, guaranteeing continuity of visualized descriptions given in typical narratives at the origin of requirements specification, in the early phases of any information system development.

As background for our conceptual representation and for the purpose of a self-contained paper, the next section reviews the general features of the model to be utilized as “a ground fabric at the base of the design.” The model has been used in many applications⁽¹²⁻¹⁵⁾, but the explanation and example presented here are new contributions.

2. Flowthing model

The Flowthing Model (FM) is a map of conceptual movement, in stages (e.g., sequential events in circulation, including flow/transportation/ communication), processing (e.g., digestion), and creation of things that are called *flowthings*. Goods, people, ideas, data, information, money, and capital that flow among *spheres* (e.g., places, organizations, machines, and cultures) are flowthings. Fig. 3 gives a general plan of a flow system, with five “stages” shown: Transfer, Receive, Process, Create, and Release. Each *stage* has its own vocabulary:

Create: generate, appear (in the scene), produce, make, ...

Transfer: transport, communicate, send, transmit, ...

Fig. 3. Flowsystem

Process: Millions of English verbs that change the form of a flowthing without creating a new one.

Fig. 4 illustrates a sphere with two flow systems. A flowsystem here is a schematic model of movement of flowthings through a sphere. The resultant schema expresses a *holistic* view of traffic and flow patterns and can be used to determine the character of the structure of the entire system.

Still, such a schema as a flowsystem “is a possibility of fact – it is not the fact itself”⁽¹⁶⁾, where a certain instance is mapped to a subdiagram of the flowsystem. For example, assume that *John sends an email to Mary, who replies by email* (a certain instance). Fig. 5 is an instance of the two flowsystems in Fig. 4 that shows this interaction. First, John creates (circle 1) an email, which is released (2) and transferred (3) to the email flowsystem of Mary.

Fig. 4. Two flowsystems.

Fig. 5. Stream of flow of email and its reply email between two flowsystems.

In Mary’s sphere, the email is taken as an input transfer (4), received (5), and processed (6). This processing (i.e., reading) *triggers* (dashed arrow, circle 7) the creation of a reply email (8) that is released (9) and transferred (10) to John. The reply email is taken as an input transfer (11), received (12), and processed (13).

FM represents a web of interrelated *flows* that cross boundaries of intersecting and nested *spheres*. A flowthing is defined as “what is created, released, transferred, arrived, accepted, and processed” while flowing within and among *spheres*. It has a permanent identity but a transient form, e.g., the same news translated into different languages. A *flowsystem* constrains the trajectory of flow of flowthings. A particular flowsystem is the space/time context for *happenings* (e.g., received, released) and existence of flowthings. To simplify the FM diagram, the interior of the stages of Transfer (input and output) and Receive (arrival and acceptance) are, most of the time, not shown.

A flowthing can be in *one and only one* of the FM stages at a time: Transfer, Receive, Process, Create, or Release, analogous to molecules of water being in one of three states in Earth’s atmosphere: solid, liquid, or gas. A *stage* here is a “transmigration field” of the flowthing that is created, processed, released, transferred, arrives, and is received. A “natural” and irreversible flow is assumed, e.g., released flowthings flow only to Transfer. In an example not depicted in Fig. 5, the possibility exists that the carrier DHL, because of unusual circumstance, e.g., a natural disaster, is not able to transfer released parcels abandoned in an airport; hence the company moves these released parcels “backward,” against the flow, to be returned to customers.

The exclusiveness of FM stages (i.e., a flowthing cannot be in two stages simultaneously) indicates synchronized change of the flowthing, e.g., a flowthing cannot be *changed* in form and sphere simultaneously. This is a basic systematic property of flowthings. Note the generality of the notion of flow in FM. For example, *creation of* a flowthing is a *flow* (from *nonexistence*, i.e., not currently existing in the system, to *existence*, i.e., first appearance in the system).

Initialization, stopping, and continuing of flows occur through *triggering*, a control mechanism. In FM description, it is the only linkage among elements besides flow and is indicated by **dashed arrows**. Synchronizations (e.g., join/fork) and logic notions (e.g., and/or) can be superimposed over the basic FM depiction. Note that these mechanisms can be modeled as flowthings.

FM, as in the case of UML and SysML, can be used as a sketching, blueprinting, and programming tool⁽¹⁷⁾ for model description as well as specification and design.

3. Example that illustrates FM

In the context of studying eGovernment in the European Union, Tobias et al.⁽¹⁸⁾ had the following goal: *interoperability of processes, services, and documents* that are exchanged between organizations in the EU. They provide an approach for “deriving documents and services from current eCustoms procedures” to be embedded in a service-oriented architecture for collaborative eGovernment. Using certain component technical specifications (the name of the particular specification language is irrelevant in the context of this example), Tobias et al.⁽¹⁸⁾ developed *a conceptual framework for modeling document components in a syntax-neutral, technology-independent manner*.

Regarding eCustoms procedures, ... crucial EU control procedures are still paper based... Hence, new customs control procedures are required which can be supported much more effectively and efficiently by the use of IT. However, designing and implementing changes in customs control procedures has to take into account

technological, financial and political issues...that can only be dissolved by *process integration*... Tobias et al.⁽¹⁸⁾ (italics added, citations omitted)

We concentrate on an overall description of one application of Tobias et al.’s⁽¹⁸⁾ approach, the so-called *excise movement and control system (EMCS)*, a system that supports monitoring the movements of goods (e.g., alcohol, tobacco, and energy products) between EU member states. Specifically, the focus is on the first UML diagram used to describe this system, shown partially in Fig. 6.

In [the] example, the consignor is a beer brewery in the Netherlands, which submits a draft e-AAD to the Dutch Tax and Customs Authority (DTA). Acceptance or validation of the e-AAD is conducted by DTA as the customs office of dispatch by automatically checking the declaration data ...⁽¹⁸⁾.

The point of examining this part of the example is not to give a complete description of Tobias et al.’s⁽¹⁸⁾ diagram; rather, it is to contrast the diagramming methodology used with the corresponding FM diagramming method, as shown in Fig. 7.

Fig. 6. Movement declaration collaboration process (partial from Tobias et al.⁽¹⁸⁾)

Fig. 7. FM representation of process shown in Fig. 6

Fig. 8 shows four spheres in the system, indicated by circles 1 through 4, and their flowsystems. These spheres are flipped horizontally in comparison with Tobias et al.'s⁽¹⁸⁾ figure for the purpose of eliminating intersecting lines. Accordingly, the flow starts at circle 5 in the figure when the <<Consignor>> Beer Brewery creates an e-AAD that flows to the <<Customs Office of Dispatch>> in the Dutch Tax Administration (circle 6), where it is validated (7). This triggers (8) the creation (9) of an e-AAD with ARC that flows to the Consignor (10) and also to the <<Customs Office of Destination>> HMRC, UK (11), where it is received (12). Also, the <<Customs Office of Destination>> creates (13) a report that flows to HMRC, UK (14) where it is validated (15) (Note: The connection between this thread that starts at Create (13) and the previously described threads is not clear from Tobias et al.'s⁽¹⁸⁾ description, but we will ignore that). This triggers (16) the creation of an acknowledgement (17) that flows to Consignee (18).

Contrasting the two diagrams (Figs. 6 and 7), we observe that the generic FM stages categorize all activities, but the UML activities in Fig. 6, e.g., send, receive, evaluate, notify, ... seem, in principle, to embrace all English verbs. Thus, the activity diagram can show unlimited “types” of vertices selected from English verbs by the designer. This gives an imprecise connotation of a mixture of different meanings for different implementations of the same system. It would be like defining a graph in graph theory where the definition allows the definer to select an infinite number of node types according to personal preference. In FM there are only five types of nodes – process, create, release, transfer, and receive – and these are repeatedly used in different spheres. This simplification produces a uniform specification method for diagramming and is thus suitable for systematic modeling.

Additionally, the edges in Tobias et al.'s⁽¹⁸⁾ figure are conceptually disturbing. Examine Fig. 8, which is part of Fig. 6. Line A seems to be an e-AAD *data flow*, but it is also a *control flow*. B seems to be a Report *data flow*. C is definitely a *control flow*. D seems to be a Report *data flow*, but it is also a *control flow*. The semantics of connectivity in this diagram seem imprecise.

Fig. 8. Different interpretations of edges

By contrast, in the FM representation, different flows have their distinct streams such that the flows do not mix together, in analogy to a building blueprint where the flows of water, gas, and electricity are not combined.

4. Case study: Applying FM to an Airport

This paper reports the results of applying FM as an integration tool to a currently existing personnel information system at Kuwait International Airport, the main airport in Kuwait. Air traffic already reflects the increased economic vitality of the airport. It handled almost 8.5 million passengers in 2011 and now hosts more than 45 different regional and international carriers serving more than 80 destinations. Airfreight is also consolidated, with more than 195,000 tons passing through its cargo facilities annually. The existing terminal features three levels plus a basement, covering a total of 40,000 m². There are 10 contact aircraft stands, 40 remote stands, and additional parking spaces for cargo and VIP aircrafts.

The current information system has been developed piece-wise from subsystems over the last ten years and includes a system for issuing employee identification cards, a biometric time attendance system, an access permission system, a vacation system, and an employee ticketing system. This paper utilizes FM to draw a base on which to integrate all these system in preparation for redesigning and developing them into a single system. Because of space limitation, the paper shows the plan to integrate only two systems.

4.1 System for issuing identification cards

FM is used to describe the whole process of issuing identification cards. This usage is similar to a civil engineer who looks down on the world while airborne to visualize the natural environment in order to create maps. The resultant geospatial information reveals new insights into the to-be-developed project. Accordingly, Fig. 9 shows the FM representation of such a process built on examination of a real environment and its current information system.

An employee (circle 1) applies for an identification card by downloading (2) an application form (3) from the airport's information technology web site (4). He/she fills out the form (5) and sends it (6) to his/her department secretary. The secretary of the department processes, e.g., registers (7) the application form and sends (8) it to the secretary of the Permissions Department, who processes (9) it and transfers it (10) to the issuer in the department.

Fig. 9. Permission issuing process

The issuer processes it (11) and adds it to the next scheduled permission meeting (12). There it is approved (13) or rejected (14). Approval triggers (14) the creation of the card (15), which flows (16) all the way back to the employee. If the application is rejected, the application is marked as such, thus triggering (17) the generation (18) of a disapproved application that flows (19) all the way back to the employee.

4.2 Vacation System

The second system, used by employees to request vacations, is shown in Fig. 10. The employee applies for a vacation in a similar fashion to applying for permission, as discussed for Fig. 9.

Space limitation prevents discussion of Fig. 10 in detail, but it follows the same basic format and principles as Fig. 10. It is not difficult to note the overlap between Figures 9 and 10, implying they could be integrated in a systematic way. The problem now is to find the union of the two graphs. For example, the employee sphere could incorporate two subspheres, one for permissions and the other for vacations.

5. Conclusion

This paper has proposed a new approach to building a basic framework for initial system description. The FM approach seems to provide a viable solution to the need for

Fig. 10. Vacation system

an integrated depiction of any system upon which other stages of development can be based. FM methodology is currently also being utilized to describe physical airport systems such as aircraft landing and departure systems.

References

- (1) G. D. Bhatt and V. Grover: "Types of information technology capabilities and their role in competitive advantage: an empirical study", *Journal of Management Information Systems*, Vol. 22, No. 2, pp. 253-277, 2005.
- (2) J. Auramo, J. Kauremaa, and K. Tanskanen : "Benefits of IT in supply chain management: an explorative study of progressive companies, *International Journal of Physical Distribution & Logistics Management*, Vol. 35, No. 2, pp. 82-100, 2005.
- (3) T. Wendt, B. Brigl, and A. Winter: "Assessing the integration of information system components", *Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems*, 2005.
- (4) Edward C. S. Ku: "Functional integration and systems implementation of customer relationship management in hotel industry: a multilevel analysis", *International Journal of Information Technology and Decision Making*, Vol. 13, p. 175, 2014. DOI: 10.1142/S0219622014500084
- (5) N. Berente, B. Vandenbosch, and B. Aubert: "Information flows and business process integration", *Business Process Management Journal*, Vol. 15, No. 1, pp. 119-141, 2009.
- (6) Alexander Egyed, Integrating Architectural Views in UML, Technical Report, USC/CSE-99-TR-514, March 10, 1999, (Access Sept 2014).
<http://csse.usc.edu/csse/TECHRPTS/1999/usccse99-514/usccse99-514.pdf>
- (7) B. Nuseibeh : "Towards a framework for managing inconsistency between multiple views", *Proceedings of the Viewpoint 96: International Workshop on Multiple Perspectives in Software Development*, 1996.
- (8) J. Zhu and A. Mostafavi : "Towards a new paradigm for management of complex engineering projects: a system-of-systems framework", 2014.
- (9) G. Finance: SysML modelling language explained, 2010 (Access Sept 2014).
http://www.omg.sysml.org/SysML_Modelling_Language_explained-finance.pdf
- (10) F. O. Hansen : SysML – a modeling language for systems engineering, 2010 [slides].
<http://staff.iha.dk/foh/Foredrag/SysML-SystemEngineering-DSFD-15-03-2010.pdf>
- (11) David Chen and Walter Stroup : "General System Theory: Toward a Conceptual Framework for Science and Technology Education for All", *Journal of Science Education and Technology*, Vol. 2, No. 7, 1993.
- (12) S. Al-Fedaghi : "Flow-based enterprise process modeling," *International Journal of Database Theory Applications Compendex*, Vol. 6, No. 3, pp. 59-70, 2013.
- (13) S. Al-Fedaghi: "A method for modeling and facilitating understanding of user requirements in software development", *Journal of Next Generation Information Technology SCOPUS*, Vol. 4, No. 3, pp. 30-38, 2013.
- (14) S. Al-Fedaghi: "Pure conceptualization of computer programming instructions", *International Journal of Advanced Computer Technology SCOPUS*, Vol. 3, No. 9, pp. 302-313, 2011.
- (15) S. Al-Fedaghi and A. Alrashed: "Threat risk modeling", 2010 International Conference on Communication Software and Networks (ICCSN 2010), Singapore, 2010.
<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5437700&url=http%3A%2F%2Fieeexplore.ieee.org%2Fstamp%2Fstamp.jsp%3Ftp%3D%26arnumber%3D5437700>
- (16) Gilles Deleuze: *Cinema I: The Movement-Image*, Trans. Hugh Tomlinson and Barbara Habberjam. Minneapolis: University of Minnesota Press, 1996.
- (17) M. Fowler: *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd ed. Object Technology Series. Addison-Wesley, 2003.
- (18) Tobias Vogel, Alexander Schmidt, Alexander Lemm, and Hubert Österle: *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 3, No. 3, pp. 17-37, 2008 (Access Sept 2014). ISSN 0718-1876.
http://www.scielo.cl/scielo.php?pid=S0718-1876200800200003&script=sci_arttext