Modelling and control of engineering plant processes

Sabah Al-Fedaghi*

Computer Engineering Department,
Kuwait University,
P.O. Box 5969 Safat 13060, Kuwait
Fax: 0096524839461
Email: sabah.alfedaghi@ku.edu.kw
*Corresponding author

Abdulaziz AlQallaf

Information Technology Department,
Kuwait National Petroleum Company (KNPC),
Al Ahmadi, Kuwait
Email: ai.qallaf@knpc.com

Abstract: With increasing scale and complexity of networked applications, the concept of system of systems has recently emerged. This paper introduces a new systems engineering approach that can be utilised uniformly to represent integrated static descriptions of systems and their behaviours along with control specifications. It avoids the multiplicity of views associated with object-oriented techniques. Its ontology is simpler in that it incorporates only a few notions: things, flows, and their stages and triggering. This paper demonstrates the advantages of the proposed modelling technique by producing a high-level systematic specification (comparable to such tools as process flow diagrams) and using this model to depict the management and control of a system. To demonstrate the viability of the model, it is applied to developing a diagrammatic description of processes in a water distillation plant.

Keywords: conceptual model; schematic diagrams; process engineering; process flow diagram; engineering system high level description.

Reference to this paper should be made as follows: Al-Fedaghi, S. and AlQallaf, A. (2018) 'Modelling and control of engineering plant processes', *Int. J. Applied Systemic Studies*, Vol. 8, No. 3, pp.255–277.

Biographical notes: 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 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).

Abdulaziz AlQallaf holds a Bachelor and Master in Computer Engineering from the Department of Computer Engineering, Kuwait University. He has been working since 2015 as a Computer Engineer in the Instrument Maintenance Department, Ministry of Electricity and Water, Kuwait. His interests include computer networks, security and software engineering.

1 Introduction

According to Tuffley and Wen (2012), "processes and process models underpin the activities of every engineering discipline". Engineering *processes* are complex and often involve diverse and serious potential consequences such as damage to the environment and dangers to people. Process models are developed for such purposes as understanding inner workings, prescribing procedures to be followed, and making predictions about the modelled engineering project. These models also aim to provide an overview of mechanisms used in system operations for decision making and control.

Several modelling techniques, e.g., SDL, UML, and SysUML, have been used to represent the static and behaviouristic features of control systems. Still, a conceptual structure is needed that would integrate constituents and permit assemblage of applications from shared processes (Al-Fedaghi, 2014). According to Dori (2003),

As the inherent complexity and interdisciplinary nature of systems increases, the need for a universal modelling, engineering, and lifecycle support approach becomes ever more essential. The unnecessary complexity and software orientation of UML – the current standard language – calls for a simpler, formal, generic paradigm for systems development.

This paper proposes a *descriptive modelling methodology* for such processes that has merits in itself as a communication tool (e.g., among technical and non-technical stake holders), with a *prescriptive* aim of improving the *management and control* of engineering systems. A *descriptive* model explains how assembled processes generally work in a system, whereas the aim of a prescriptive model is to refine and improve the work of the system. The proposed approach produces a high-level systematic specification (comparable to such tools as process flow diagrams) that is used in management and control of the involved system.

To demonstrate the viability of the methodology, it is applied to developing process diagrams of a currently existing *water distillation plant*. The model is based completely on a conceptual/logical approach, in contrast to technical, hardware, and software representations, making it suitable for applications in various engineering fields.

Without loss of generality, this paper focuses on the area of *process flow diagrams*. "Visual information is the clearest way to present material and is least likely to be misinterpreted. The most effective way of communicating information about a process is through the use of flow diagrams" (Turton et al., 2009).

Many types of schematic diagrams are used in diverse engineering disciplines for proper documentation of entire systems and for control and monitoring functions. These include fluid power (hydraulic and pneumatic) schematics, heating, ventilation and air conditioning (HVAC) control logic diagrams in mechanical engineering, site plans and building layouts, borehole logs in civil engineering, process flow diagrams and piping and instrumentation diagrams in chemical and systems engineering, and circuit schematics and logic diagrams in electrical engineering.

1.1 Brief review of area of interest: process flow diagrams

While a comprehensive review of theories of modelling of engineering processes is beyond the scope of this paper, this sub-section highlights *the type of model* of interest in this paper.

A *process flow diagram* is a schematic representation of the sequence of all operations "occurring when an *object* (or material) is intentionally *changed* in any of its physical or chemical characteristics, is assembled or disassembled from another *object* or is arranged or prepared for another operation, transportation, inspection or storage" (Kolmetz et al., 2015; italics added). Notions such as *object* and *change* have generally understood meanings. The diagram typically shows the following:

- the relationship between major equipment of a facility, but does not show minor details such as piping details and designations
- tabulation process design values for the components in different operating modes
- the interconnection of process equipment and the instrumentation used to control the process (Siddiqui et al., 2015)
- facilitation of understanding of the involved processes
- a holistic view of the design of main streams of flow and operations
- a base for management and control as well as automated control
- use in such aspects as security and safety management.

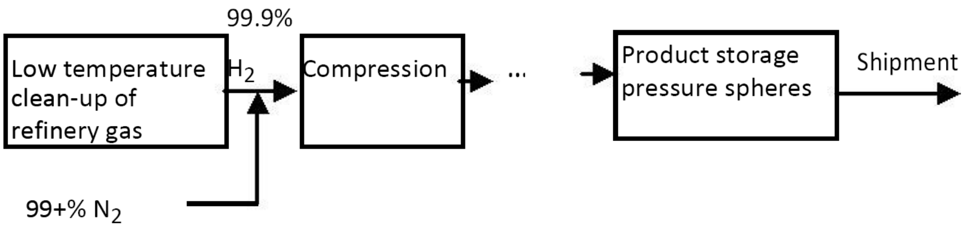
In the process industry, drawings of processes are specified according to standard symbols [e.g., *ISO DIS 10628-2: Diagrams for the Chemical and Petrochemical Industry* (ISO, 2010)] in order to make the development of flow schemes easier (Siddiqui et al., 2015). More-detailed diagrams (may be called process and instrument drawings, piping and instrumentation drawings, or flow sheet diagrams) include minor flows, control loops, and instrumentation used by technicians as well as by instrumentation, electrical, process, and safety engineers. They display control valves and valves that affect operation of a system, interconnections with other systems, bypasses, and recirculation lines.

A simple form of presentation in this context is a block diagram, in which each block represents a single piece of equipment or a stage in the process. Describing a block diagram, Turton et al. (2009) noted that:

> [The block] diagram consisted of a series of *blocks* representing different equipment or unit operations that were connected by input and output streams… However, the diagram did not include any *details of equipment within any of the blocks*. (Italics added)

With complex processes such diagrams are limited to representing the overall process, broken down into principal stages. The representation usually shows a preliminary processing concept without details. "The blocks do not describe how a given step will be achieved, but rather what is to be done" (Kolmetz et al., 2015). Block diagrams are also useful for representing a process in simplified form, but they have limited use as engineering documents (Kolmetz et al., 2015). Figure 1 shows a partial picture of this type of diagram.

Figure 1 Block flow diagram of ammonia process



Source: Redrawn, partial from Kolmetz et al. (2015)

1.2 Problem: need for a single, unifying model

One can sense the absence of *global integrated system conceptualisation* in the area of engineering processes. Chen and Stroup (1993) articulate five motivations for developing a unifying integrated framework for systems: integration, engagement with complexity, understanding change, relating macro- and micro-levels, and functioning in a human-made world.

This lack of *global integrated conceptualisation* was explicitly raised in Dori (1995), viewed at that time in the context of software engineering:

> Just as the procedural approach to software was not necessarily the most adequate one, neither was the 'pure' OO [object-oriented] approach, which puts objects as the sole 'first class' citizens, with 'methods' being their second-class subordinate procedures.

In the object-oriented paradigm, an object is a key concept that denotes real-world objects. Objects have state and behaviour; "while object-orientation has been a successful software design approach, the world is not inherently object-oriented" (Liu and Gluch, 2004). "The idea of an object is a programming idea, and it doesn't fit most of the individuals in the world very well" (Jackson, 1995). Accordingly, object-process methodology (OPM) (Dori, 2003) is a 'more natural' (Liu and Gluch, 2004) systems engineering approach to modelling real world problems.

Accordingly, OPM has been developed as a comprehensive approach to systems engineering that integrates function, structure, and behaviour in a single, unifying model (Dori, 2002). OPM is specified as ISO/PAS 19450 (ISO/PAS, 2015). This paper examines the same problem and develops an alternative methodology to OPM.

1.3 Contribution

The underlying thesis of the paper is that development of a *global integrated conceptualisation* 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" (Al-Fedaghi and Haider, 2015).

The proposed methodology, called the FM model, captures the static and dynamic aspects of a system in a *single integrated* view (Reinhartz-Berger and Dori, 2004). The model can serve as a ground fabric for various types of engineering applications; however, in this paper, it is applied to model operations in a water distillation plant.

We claim that the FM model has certain advantages over other models of engineering processes such as OPM. At this point in the development stage, such a claim can be supported by constructing an FM model of a process and comparing it side by side with a different model of the same process.

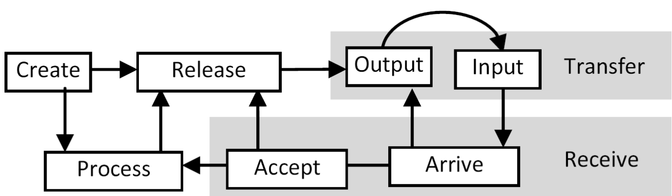
For the sake of a self-contained paper, the basic notions in FM are briefly described in the next section. The model has been utilised in software engineering (e.g., Al-Fedaghi, 2014, 2016a, 2016b, 2017a, 2017b, 2017c; Al-Fedaghi and AlQallaf, 2017). Section 3 gives an example illustrating the features of the model. Section 4 demonstrates additional features of the FM model, including its use as the basis for appropriate understanding of notions commonly known by such terms as *events* and *functions*. Section 5 applies FM to a case study of an actual engineering plant. Section 6 discusses some conclusions of applying the FM model to the case study.

2 FM model

> One key insight has been the shift from the understanding of *a process* as the *transformation* of materials from inputs to outputs to the view of a process as a flow of materials through a sequence of steps or operations … Anybody having encountered [a] process will know that there is a plethora of flows feeding the process. Some flows are easily identified, such as materials flow, whilst others are less obvious, such as tool availability. Some are material while others are non-material, such as flows of information, directives, approvals and the weather. But all are mandatory for the identification and modelling of a sound process. (Henrich et al., 2007, italics added)

In a broader sense, a process *not only transforms things* (objects) but also *creates*, processes, *releases*, *transfers*, and *receives* them. The FM ontology includes *things*, these five 'states', and *flow*, all wrapped up in (abstract) flow machines (see Figure 2). *Things* are what can be created, processed (transformed or changed), released, transferred, and received where at any specific point in time, a thing will be in exactly one of these 'states'.

Figure 2 Flow machine



Things can exist, e.g., a product is created, released, transferred to consumers, received, and processed (consumed), and *things* can also happen, e.g., an event that is created, processed (runs its course), released, and transferred. *Machines* can themselves be *things* that are created, processed, and so on. A *structure* (baseline description) of a system is a machine, and its behaviour is an 'event-ised' structure, as will be demonstrated in this paper.

The machine includes stages or 'states of a thing' that can be described as follows:

- *Arrive*: a thing reaches a new machine.
- *Accepted*: a thing is permitted to enter the system. If arriving things are also always accepted, arrive and accept can be combined as a single *received* stage.
- *Processed*: the thing passes through some kind of transformation that changes its form but not its identity (e.g., the same news translated into different languages).
- *Released*: a thing is marked as ready to be transferred (e.g., airline passengers cleared and waiting to board).
- *Created*: a new thing originates (is created) in the system (e.g., a data-mining program generates the conclusion that an application is rejected as input data).
- *Transferred*: the thing is transported somewhere outside the machine (e.g., packets reaching ports in a router, but still not in the arrival buffer).

These stages are mutually exclusive; i.e., a thing in the *process* stage cannot be in the *created* stage or the *released* stage at the same time. An additional stage of *stored* can be added to any FM model to represent the storage of things; however, storage is a generic, nonexclusive stage, because there can be *stored processed things*, *stored created things*, and so on.

Figure 2 shows the structure of a machine and its internal flows with its stages and transactions (arrows) among them. A machine may not need to include all the stages; for example, an archiving system might use only the stages arrive, accept, and release. Multiple systems captured by FM can interact with each other by *triggering* events related to one another in their spheres and stages.

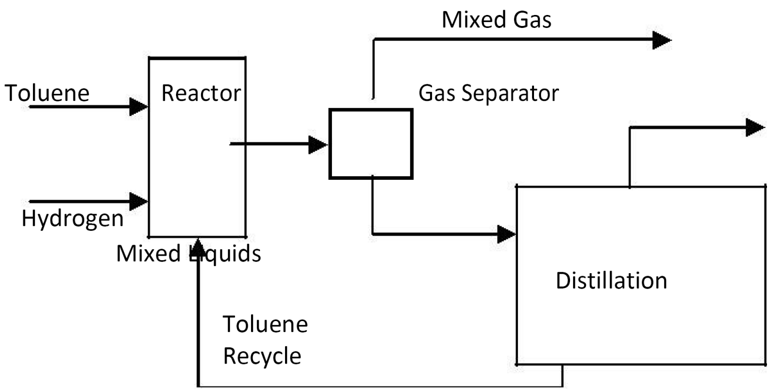
Spheres and subspheres are the environments of the flow of things. A sphere can have multiple machines in its construction, if needed. It can be an entity (e.g., a hospital and the departments within it; a person or class of persons such as nurses; a transistor; a logic gate, channel, a wire). A machine is a subsphere that embodies the flow; it itself has no subsphere.

Triggering is an instrument of activation (denoted by a dashed arrow). The mechanism of triggering can control the movement of things in the system; e.g., if a thing satisfies some condition, it is allowed to flow to *release*. A flow is said to be triggered if it is created by another flow (e.g., a flow of electricity triggers a flow of heat) or is activated when a condition in the flow is satisfied (e.g., processing of records x and y triggers the creation of record z in a machine of records).

3 Example

Turton et al. (2012) give an example of a block flow diagram of the production of benzene (see Figure 3). Toluene and hydrogen are sent to a reactor, and the effluent is sent to a gas separator where the non-condensable gases are discharged from the system. The bottom of the separator provides a liquid feed to a still where the lighter benzene gas is collected as the distillate, and the bottom toluene draw is recycled back into the reactor (Pinkerton et al., 2014).

Figure 3 Block flow process diagram for the production of benzene



Source: Redrawn, partial from Pinkerton et al. (2014), taken from Turton et al. (2012)

3.1 FM baseline description

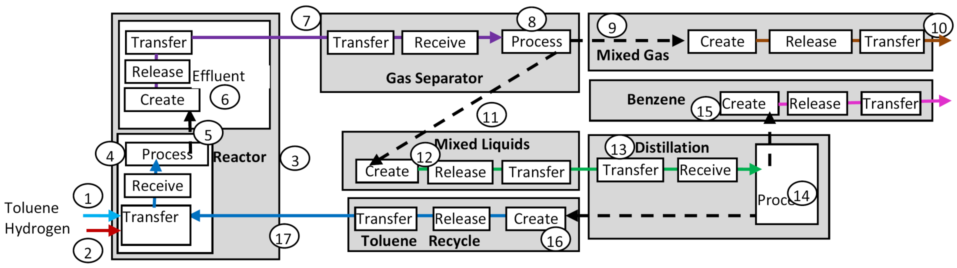
Figure 4 shows the FM representation of this example. The toluene and hydrogen (circles 1 and 2 in the figure) flow to the reactor (3), where they are mixed and processed (4) to trigger (5) the creation (6) of the effluent that flows to the gas separator (7). There, it is processed (8) to

- trigger (9) the release of mixed gas (10)
- trigger (11) the creation of mixed liquid (12) that flows to distillation (13).

There, it is processed (14) to

- produce (create) benzene (15) that is released
- produce (create) toluene recycle (16) that flows to the reactor (17).

Figure 4 FM representation of block flow process diagram for the production of benzene (see online version for colours)



Note that since the authors of the present paper are not expert in this field, there is a possibility of technical inaccuracy in Figure 4; nevertheless, the point here is not to explain the production of benzene; rather, it is to assess its representation, the type of diagram of this process. If this representation is technically incorrect, it can be corrected to produce a representation similar to Figure 4.

3.2 *Event-ised description*

Figure 4 is a static model in the sense that it is the 'space region' for *happenings*. "It is a possibility of fact – it is not the fact itself" (Deleuze, 1996) that is actualised by a certain instance (event) that 'activates' a sub-diagram (e.g., the sub-diagram depicting the Reactor is activated by the actual flow of toluene and hydrogen at a certain time). The order of the relationships (among sub-diagrams, e.g., gas separator *before* mixed gas) in Figure 4 is based entirely on logical relationships.

3.3 *Behaviour: event-ised description*

The next level in FM modelling is to identify all types of *events*. Events in FM are treated as *things* (i.e., what can be created, processed, released …). Events are identified by their *machine*, *time, region,* and other aspects not discussed here. For example, Figure 5 shows the event of *producing benzene* in a specific time as a separate phenomenon occurring after distillation. It can be noticed that it contains many elementary events, e.g., creation, release, and transfer; however, we select this event as the 'meaningful' event, just as the French Revolution is conceived as a historic event even though it comprises numerous small events. Also, the event is conceptualised as one discrete event (within a period of time that can be be made smaller *infinitesimally*), even though its creation is a continuous creation of Benzene.

Figure 5 The event: producing benzene (see online version for colours)

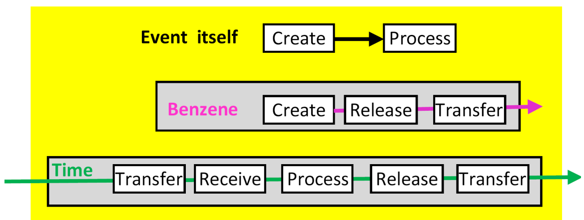
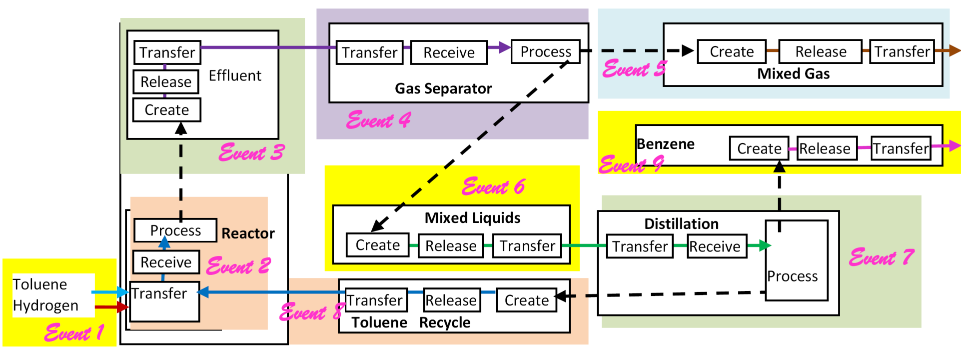


Figure 6 Events (see online version for colours)



Accordingly, *event-ising* the static baseline description refers to identifying all types of 'meaningful' events, as shown in Figure 6. Note that, for simplicity, only the regions of events are indicated in the figure while their machines and time machines are ignored.

3.4 Control

Let each event be denoted by V_1, then the dynamic behaviour (execution) of the system can be described as: *Repeatedly* $[V_1, V_2, V_3, V_4, \{V_5, V_6\}, V_7, \{V_8, V_9\}]$

The curly brackets indicate possible parallelism. All types of constraints and measurements can be assigned to different phases in the execution of a system according to the events indicated. Suppose that the hardware includes two gas separators, V_4', V_4'', then another possible specification of the dynamic behaviour is:

$$\textit{Repeatedly}\left[V_1, V_2, V_3, \{V_4', V_4''\}, \{V_5, V_6\}, V_7, \{V_8, V_9\}\right]$$

A constraint in this case: if $(V_4'$ OR $V_4'')$ fails then sound alarm.

Note that a failure is an event.

4 FM methodology: illustration of some notions

This section has dual objectives: first, it shows how use of FM as a methodology (systematic, theoretical thesis of flow), as a way of thinking, and as a diagrammatic modelling language can provide the basis for "a holistic approach that enables modeling the system structural, behavioral, functional, and architectural aspects in a single coherent framework" (Reinhartz-Berger and Dori, 2004). Second, the section shows how FM can provide the basis for appropriate understanding of what is commonly referred to in terms such as *function*, *state*, *event*, and *process*.

In some of the current process models,

> [There is] a lack of consensus within engineering and design methodology about how to define key concepts such as *function* and *behaviour* … The use of different meanings for key concepts becomes problematic only when one compares and integrates different models." (Vermaas and Dorst, 2007; italics added).

Many such notions can be explained more richly using FM modelling. Because of space limitations, only some aspects of the concepts of *event* and *function* will be discussed.

4.1 Illustration of the notion of system

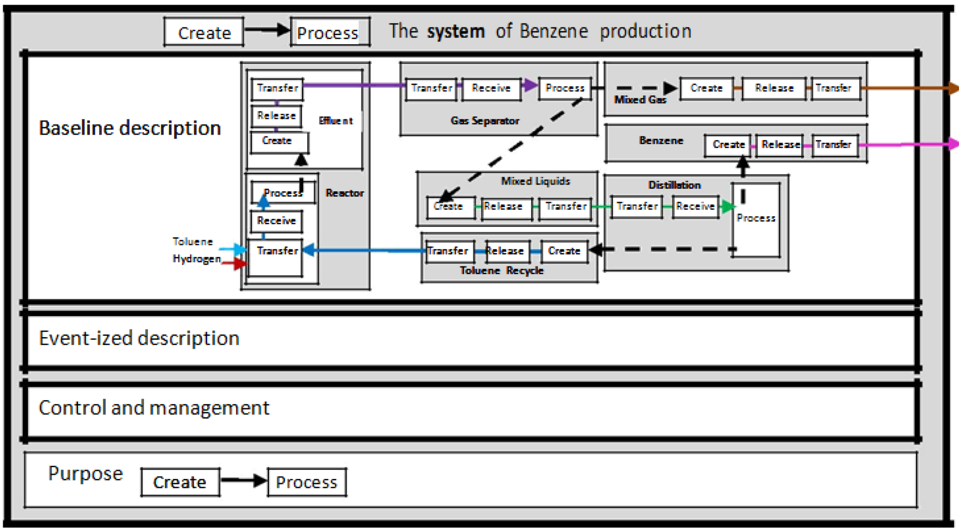
A *system* is typically described as "a set of elements that interact with one another in an organized or interrelated fashion toward a common purpose that cannot be achieved by any of the elements alone or by all of the elements without the underlying organization" (SMC, 2013).

Instead of a 'set of elements', (abstract) machines form the 'operational' part of the system, and instead of 'interact', we use the terms *transfer* (input/output) and *triggering* among machines. In FM methodology, the machine is the simplest type of system. For example, *a system is a thing that comprises flow machines that create, process, release,* *receive, and transfer things (may be sub-systems) to/from each other and across sub-systems, triggering mechanisms, and*

- a behaviour described in terms of events
- a control that manages the events, e.g., order, redundancy, parallelism, etc.
- purpose.

For example, Figure 7 shows the system of Benzene production discussed in the previous section.

Figure 7 The system of benzene production (see online version for colours)



We will not elaborate on the notion of the *purpose* of a system. It is a *thing* in the sphere of a system.

> When creating a new system or modifying an existing one, it is done in order that the resultant system does something 'useful'. The reason useful is in quotation marks is that 'usefulness' of a system depends upon the viewpoint of the observer. The purpose of a system is a property of the whole and not in any of the components. (Burge, 2015)

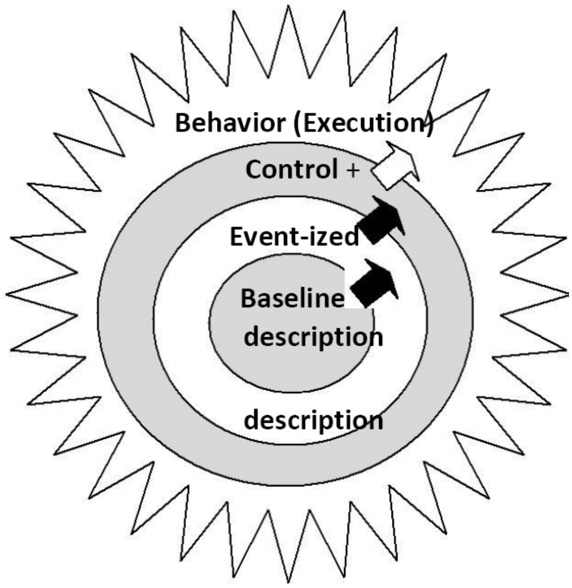
The notion of function discussed in the next sub-section is related to this topic.

According to Burge (2015), "When we consider the purpose of a system, logic and the laws of the universe dictate that it requires the achievement of lower level purposes". In the context of FM modeling, and based on this view of a system, this paper will describe:

- a baseline description of process flow
- a specification of its behaviour
- control of such behaviour.

Figure 8 shows a general view of the proposed FM modelling.

Figure 8 General view of the utilisation of the proposed model in this paper at different levels of systems specification and control



Notes: Black arrows denote 'developed from' and the white arrow denotes 'used in managing'. 'Control +' indicates additional operations such as monitoring, security, and safety specifications.

4.2 Illustration of the notion of function

Consider the causal-role theory of Cummins in which "functions are understood as causal roles subsystems play in larger systems or, more precisely, in capacities of these larger systems" (Vermaas and Dorst, 2007).

> Take, for instance, a vessel with the capacity not to rupture due to a safety valve that is part of the vessel. The causal role the valve plays in this capacity of the vessel not to rupture, is to open if the pressure in the vessel exceeds a certain value, and this role is then on Cummins' theory the valve's function. (Vermaas and Dorst, 2007)

According to this theory, the *function* of the valve is the valve's *capacity* to open if the pressure exceeds a certain value.

> For artefacts the capacities refer to physical *dispositions*, that is, to the way in which artefacts physically react to given physical circumstances. The standard example of a physical disposition is 'solubility': something has the disposition 'soluble in water' if it dissolves when put in water. (Vermaas and Dorst, 2007; italics added)

While this seems to have a deep philosophical base, for engineering purposes, the FM approach presents an alternative conceptualisation.

Figure 9 Vessel without a valve (see online version for colours)

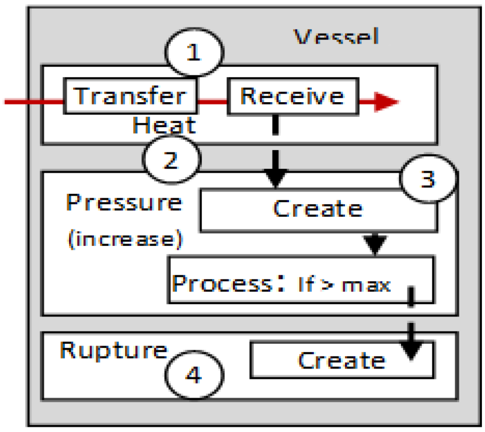
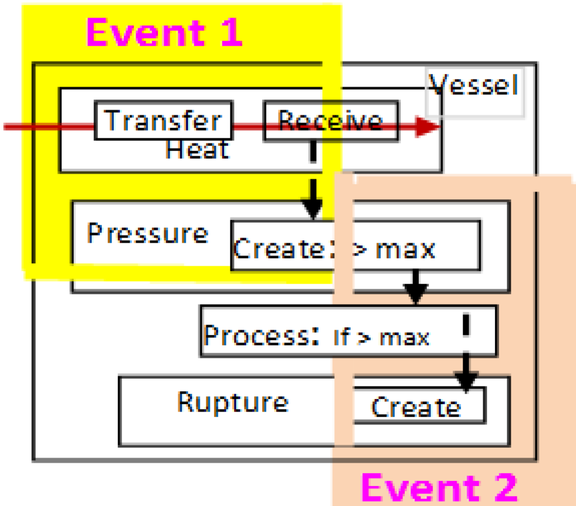


Figure 9 shows the FM representation of the vessel example where, to complete the description, it is assumed that the flow of heat is the cause of the increase in pressure. According to the figure, receiving more heat (circle 1) increases pressure (2) such that when a certain level is reached (3), a rupture is generated (4). At the events level, two relevant events can be identified (Figure 10):

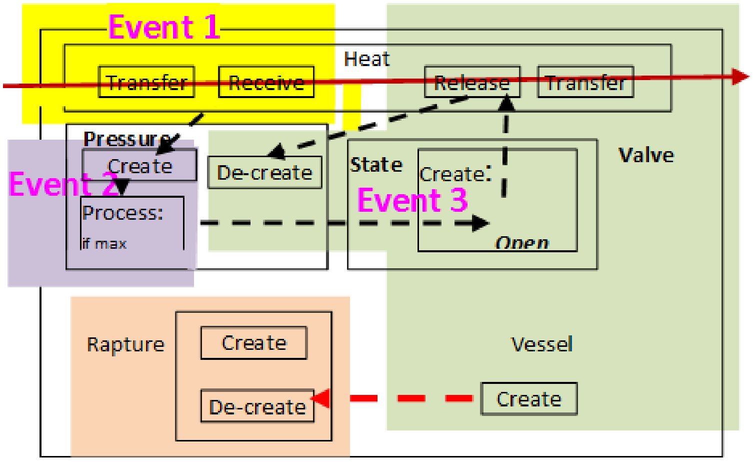
- receiving heat that results in increase in pressure
- the pressure reaching more than the maximum, thus generating a rupture.

Figure 10 Events without valve (see online version for colours)



Adding a valve produces an additional possible third event (Figure 11) of preventing a rupture from occurring.

Figure 11 Functions of the valve (see online version for colours)



4.3 *Illustration of the notion of complex events*

According to Luckham and Schulte (2011), an *event* is "anything that happens, or is contemplated as happening". A *complex event* is an event that represents a set of other events. Real-world events are observed by sensors, which translate them to simple event objects (Rinne et al., 2013).

> We may have sensors measuring the water level and flow in different parts of a network of rivers and lakes. That information combined with a weather forecast for heavy rain could be used to derive a flood warning, which in this case would be an abstract complex event object. (Rinne et al., 2013).

Figure 12 shows the FM representation of this weather forecast example. We have sensors (1) that create data about water level and flow in rivers and lakes (2); these data flow (3) to an analyst (4), who processes the data (5) to generate a flood warning (6).

Figure 12 Events in the weather forecast example

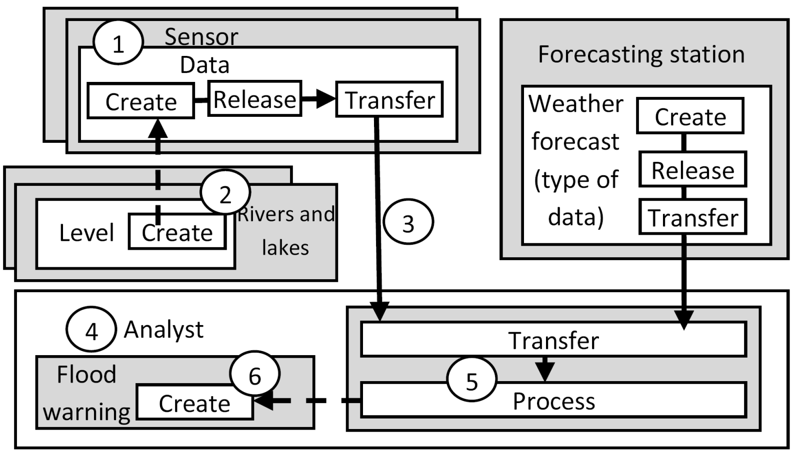


Figure 13 Event-ised weather forecast example (see online version for colours)

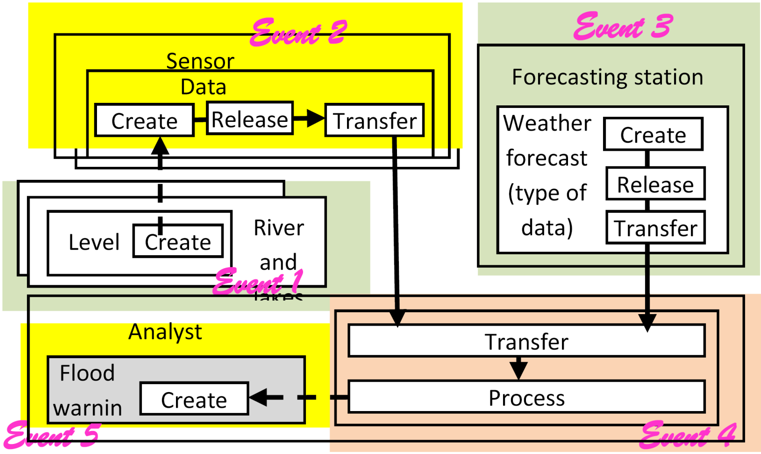


Figure 13 shows the event-ised description. The sequence of events can be expressed as V_1, {V_2, V_3}, V_4, V_5, assuming that V_2 and V_3 are generated simultaneously. Suppose the network is a a river and two lakes, each with one sensor, then the sequence is {V_1', V_1'', V_1''', V_2', V_2', V_2'''}, V_3, V_4, V_5, *etc.* There is also the possibility of more than one forecasting station; accordingly, the *complexity* of events can be reflected in the complexity of the FM diagram. It seems that the FM representation contributes to expressing the notion of complex event in more elaborate fashion.

5 Case study: water distillation plant

This section reports the results of applying FM as a modelling tool to a current system in Kuwait, the Shuaiba South Power and Water Production Station (SSPWPS). Because of continual industrial development and construction over the last several decades, a serious need arose for more electrical power and potable water. Therefore, it was decided to establish a new plant in Shuaiba called the South Shuaiba Plant, comprising six generators, each with a capacity of 134 megawatts. The first generator began work in 1970, making this water station one of the oldest in Kuwait. Total compound electrical power for the plant reached 804 megawatts, and production reached 3,032 million kilowatts per hour in 1998 (MEW, 2017).

The power plant operation includes feeding the steam from the boilers, operating at a pressure of 1250 PSI. The distillation unit comprises six distillers, each with a capacity of 6 million imperial gallons per day (MIGD), for a total of 36 MIGD. Optimum production averages 5.5 MIGD from each distiller. Distillation works through multi-stage flashing (MSF). Water from the sea passes through the screening system, where sediments and brine are filtered through mesh screens and brine pumps. This feed is given to the MSF chamber where the water and steam condense through a conduction process. This condensed distillate water is collected in a tray, and the feed is transferred to the water network by discharge pumps from each distiller. The steam is taken from boiler drum to distillation for the MSF process. The heat exchangers are used for the condensation process, creating a vacuum bed in the MSF. The distillation plant uses techniques to serve potable water to the water networks.

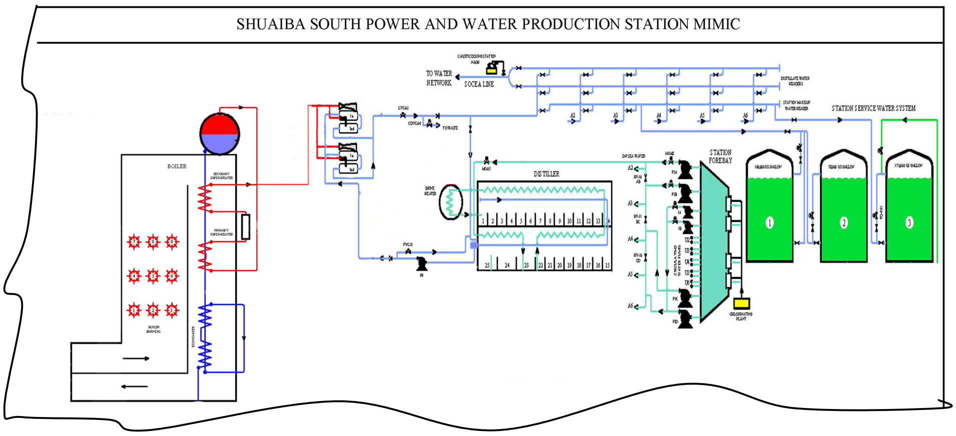
FM will be utilised to describe a portion of the water production process in a diagram. Such a description can be utilised as a base for different aspects of control, management, maintenance, and monitoring of SSPWPS. It can answer such questions as what does SSPWPS or portions of it do, how does it work, and how was it built. Building the FM representation is a venture to discover the fundamental principles underlying and enabling each part of SSPWPS through the systematic investigation of its structure, function, and operation.

In addition to these possible uses of FM diagrams in communication, control, and monitoring functions, more reasons can be offered to support the claim that such a project, in general, has practical benefits:

- Documentation of technological systems is not always a high priority, and lack of supporting records is not uncommon. Accordingly, the FM project is an opportunity to develop full documentation.
- The subsequent diagrams can be used as a pedagogical tool, especially with newly employed engineers.

A mimic of the modelled portion is shown in Figure 14.

Figure 14 Portion of SSPWPS to be modelled (see online version for colours)



5.1 *FM baseline description*

Figure 15 shows the FM representation of the portions of the plant shown in Figure 14. The process of water distillation starts by mixing Chlorine (circle 1 in the figure) with sea water (2) to produce Chlorinated water (3).

- Flow of Chlorine (1) to Chlorinated water (3): Chlorine flows in the Common header pipe (4), which is controlled by a control valve (5). The valve has two states, OPEN and CLOSED (6). When the state of the valve is OPEN, it triggers (7) the flow of Chlorine.

  The Chlorine flows (8) to four secondary pipes (9), each with its own valve (note that, for illustrative purposes, pipe machines are drawn in the shape of cylinders). Finally, the Chlorine flows to mix with the sea water (10).

- *Flow of sea water (2) to chlorinated water (3):* sea water (2) flows through a waterway (11) with a system of mesh filters to remove debris (12). This water (13) mixes with the chlorine (10) to produce chlorinated water (3).

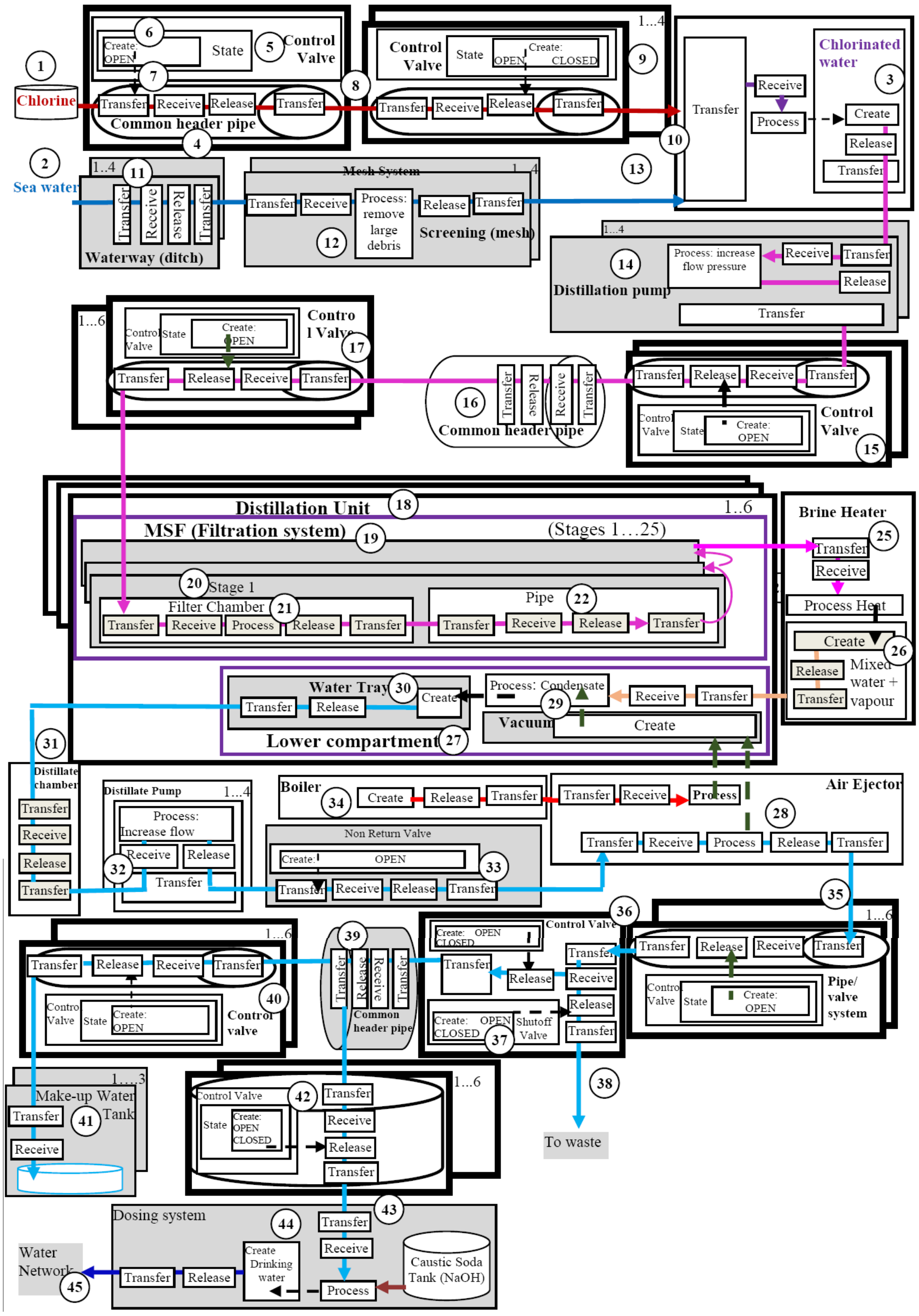
- *Flow of chlorinated water (3) to distilled water (31):* the chlorinated water (3, top right) flows through four distillation pumps (14) that increase the flow to a common header pipe (16) via a pipe/valve system (15), then to another pipe/valve system (17), and into the distillation unit (18). This unit includes the MSF unit (19) that forms the filtration system. There are six MSFs, each with 25 stages. The MSF (19) is the upper compartment of a distillation unit (18). Water flows in stage 1 (20) from the filter chamber (21) through a pipe (22) to other stages (23) until the final stage of the MSF (24), to flow to the brine heater (25). The brine heater is used to heat the water to trigger the creation (26) of mixed water and vapour. The mix flows to a lower compartment (27) of the distillation unit. The air ejector (28) is used to create a vacuum in the lower compartment so the water vapour condenses (29) and collects in the water tray (30) and then flows to be collected in the Distillate chamber (31).

- *Flow of distilled water (31) to waste (38):* the water then flows to the distillation pump (32) to increase its pressure to pass through the non-return valve (33) to arrive at the air ejector (28). There, the steam arriving from the boiler (34) and the water coming from the non-return valve (33) together cause the creation of a vacuum in the lower chamber through a heat exchange process. As mentioned previously, this vacuum triggers the condensation process of mixing water and vapour into water (30).

  Next, the heated water flows from the air ejector through a pipe valve system (35) and arrives at a control valve (36) and a shutoff valve (37) that is used to dump the water to waste (38) in case of high conductivity, making the water unsuitable for further processing. Other than this case the water flows normally to a common header pipe (39).

- *Flow of distilled water (31) to water network (45):* the water flows through a pipe valve system (40) heading toward the makeup water tanks (41) to be used later in the boiler (35). In parallel, water flows from the Common header pipe (39) to a pipe/valve system (42) that either permits or blocks its flow to the dosing system (43). There, caustic soda is added to neutralise the water and make it suitable for human consumption (44). Finally the water is transferred to the water distribution network (45).

Figure 15 FM representation of the water production portion of the plant (see online version for colours)



5.2 Event-ised description

Because of space limitations we event-ise only the top portion of Figure 15 in Figure 16, which shows four types of meaningful events:

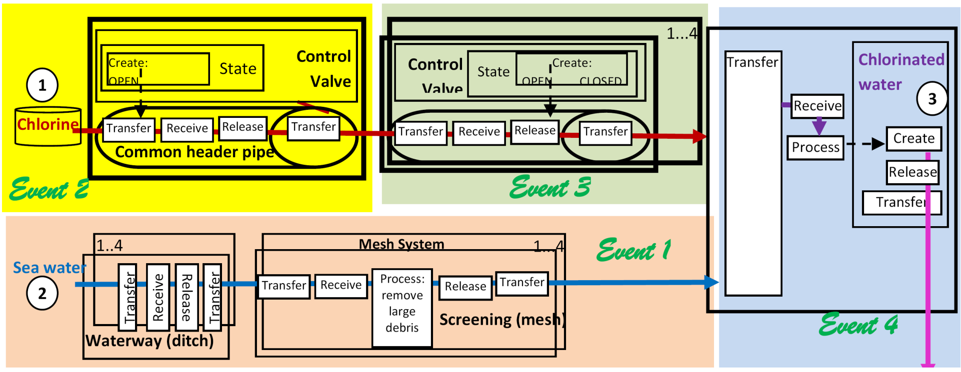
V_1 flow of sea water to the unit where it mixes with the chlorine

V_2 flow of chlorine in the common header pipe

V_3 flow of chlorine in the four pipes that follow the common header pipe

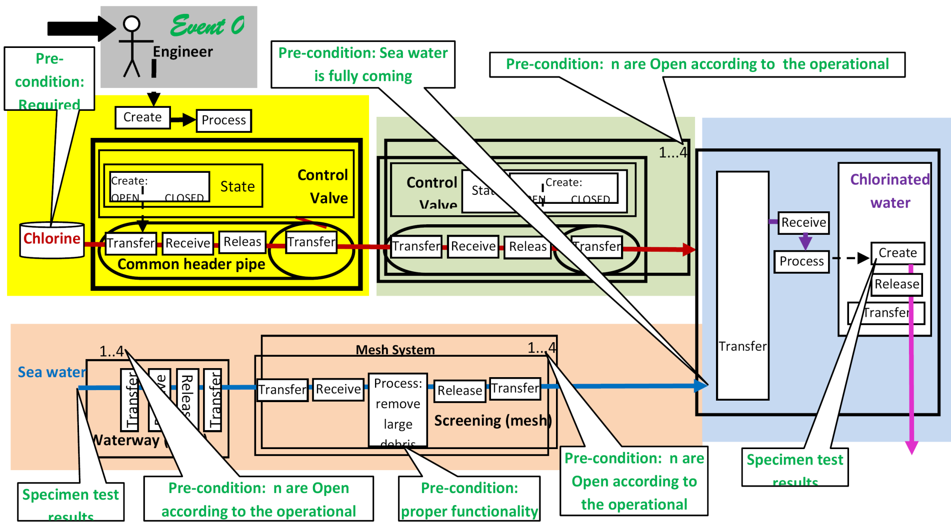
V_4 mixing of sea water and chlorine water.

Figure 16 Event-ised portion of Figure 15 (see online version for colours)



It is possible that not all four pipes of V_3 are open in an instance of execution; accordingly, V_3 can be viewed as four events: V_3', V_3'', V_3''' and V_3'''' .

Figure 17 Utilisation of the event-ised diagram for specifying starting conditions (see online version for colours)



The sequence of events in this part of the plant:

$$V_1, V_2, (subset\ of\ V_3), V_4$$

Various aspects can be applied for controlling and managing the resulting flow, e.g., operational, informational, safety and security, etc. For example, Figure 17 shows conditions for starting the system. We assume that event V_0 is this starting event, initiated by an engineer who opens the control valve that allows the flow of chlorine (noted by the black arrow in the top corner of the figure). Such an event must conform to several constraints highlighted in the figure:

- event 1 has already occurred; that is, sea water is flowing to the mixture chamber (circle A in the figure)
- the chlorine is at the correct level (B)
- the number of open pipes is in accordance with operational policy, and similar conditions.

For control purposes, various types of sensors can be depicted along the streams of flow in FM diagrams. These would be connected to a computer system to create an online control system. An automated monitoring and control system can be developed to collect different types of data, as shown for the common header pipe in Figure 17. This would involve:

- the pipe itself: data collected by sensors about the pipes such as physical conditions and similar information
- the chlorine as a flow thing: data from each stage of the flow such as rate of flow, concentration, etc.
- the state of the valve.

Figure 18 Possible control interface for the common header pipe (see online version for colours)

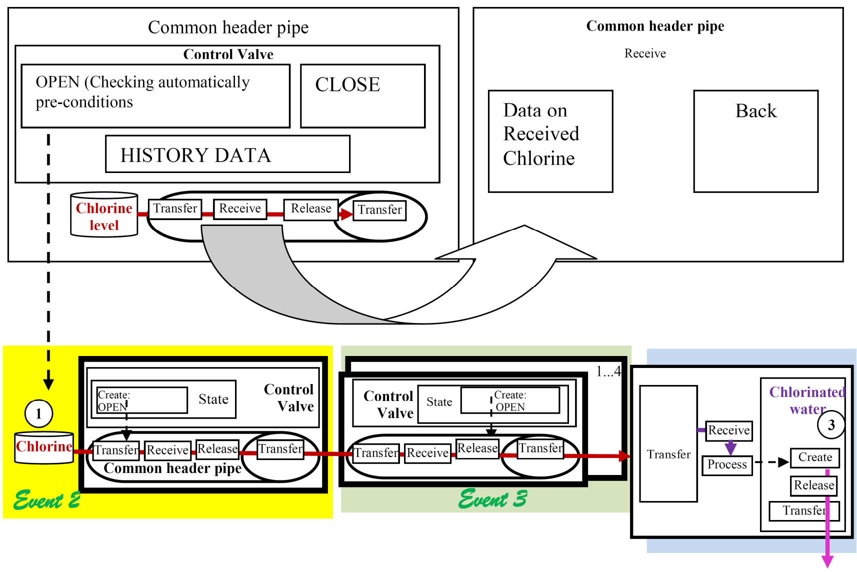
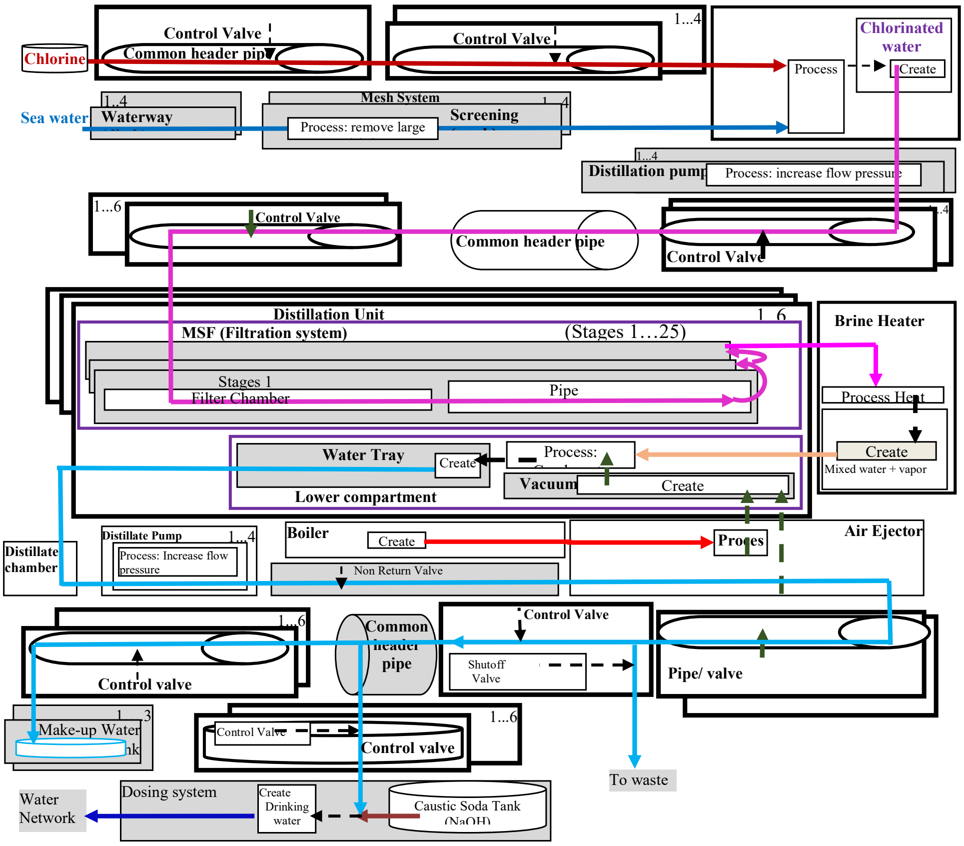


Figure 18 shows a partial view of the FM diagram, with the engineer in Figure 17 now replaced by direct connection to the control system. The figure shows a sample screen in this system that is designed to reflect the flow in the common header pipe. As usual in such an interface, each icon may lead to mare details. For example, the receive icon displays a screen that provides detailed data on the chlorine in the common header pipe taken directly from the relevant sensors.

Figure 19 Simplification of Figure 15 (see online version for colours)



6 Conclusions

The FM model is a new systems engineering approach that can be utilised uniformly to represent integrated static descriptions of systems and their behaviour to develop a control and management specification. It avoids the multiplicity of views associated with object-oriented techniques. Its ontology is simple in that it incorporates few notions: things, flows, and their stages and triggering.

Accordingly, the FM model addresses the challenges in the creation of integrated tools and techniques for better analysis and management of engineering projects by incorporating the following features:

- integration is achieved through a conceptual apparatus that assimilates multi-domain activities and interactions, e.g., physical, informational, technical
- common understanding is facilitated through use of one language for specification and control as well as communication
- complexity is resolved through simple, uniform notations applied across macro- and micro-levels of detail.

With regard to complexity, it can be argued that the FM diagram seems complex in comparison with such diagrams as UML and SysML. According to Reinhartz-Berger and Dori (2004),

Complexity management aims at balancing the trade-off between two conflicting requirements: completeness and clarity. Completeness requires that the system details be stipulated to the fullest extent possible, while the need for clarity imposes an upper limit on the level of complexity.

The alleged complexity of FM arises from more complete specification resulting from repeated application of the five stages of the flow machines. It is possible to apply granularity levels, refinement, and zooming as in a digital map to reduce the appearance of complexity seen n Figure 15. Additionally, this FM diagram can be simplified by omitting a few details such as Transfer, Receive, and Release stages in addition to the valve machines; see Figure 19 as a simplification of Figure 15. Still, all phases of the system exist because simplification is based on a complete description.

This paper has demonstrated that FM presents a viable methodology that can be used in engineering systems. Future research aims at implementing a small system based on FM to produce some evaluating metrics.

References

Al-Fedaghi S. and AlQallaf, A. (2017) 'conceptual modeling for control of a physical engineering plant: a case study', *International Journal of Computer Science and Information Security*, Vol. 15, No. 8, pp.158–169.

Al-Fedaghi, S. (2014) 'Systems design: SysML vs. Flowthing modeling', *International Journal of Software Engineering and Its Applications*, Vol. 8, No. 1, pp.355–370.

Al-Fedaghi, S. (2016a) 'Conceptual modeling in simulation: a representation that assimilates events', *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 10, pp.281–289.

Al-Fedaghi, S. (2016b) 'Toward a philosophy of data for database systems design', *International Journal of Database Theory and Application*, Vol. 9, No. 10, pp.47–62.

Al-Fedaghi, S. (2017a) 'Context-aware software systems: toward a diagrammatic modeling foundation', *Journal of Theoretical and Applied Information Technology*, Vol. 95, No. 4, pp.936–947.

Al-Fedaghi, S. (2017b) 'Flow-based provenance', *Informing Science*, Vol. 20, pp.19–36.

Al-Fedaghi, S. (2017c) 'Modeling events as machines', *International Journal of Computer Science and Information Security*, Vol. 15, No. 4, pp.174–180.

Burge, S. (2015) *System Purpose, Systems Thinking, Training and Consultancy in Systems Design and Process Improvement* [online] https://www.burgehugheswalsh.co.uk/Uploaded/1/Documents/System-Purpose.pdf.

Chen, D. and Stroup, W. (1993) 'General system theory: toward a conceptual framework for science and technology education for all', *Journal of Science Education and Technology*, Vol. 2, No. 3, pp.447–459.

Deleuze, G. (1996) *Cinema I: The Movement-Image*, in Tomlinson, H. and Habberjam, B. (Trans.), University of Minnesota Press, Minneapolis.

Dori, D. (1995) 'Object-process analysis: maintaining the balance between system structure and behavior', *Journal of Logic and Computation*, Vol. 5, No. 2, pp.227–249, doi:10.1093/logcom/5.2.227.

Dori, D. (2002) *Object-Process Methodology: A Holistic Systems Paradigm*, Springer-Verlag, Berlin, ISBN 3540654712, doi:10.1007/978-3-642-56209-9.

Dori, D. (2003) 'Object-process methodology and its application to the visual semantic web', *Pre-Conference*, Tutorial PT1 ER-2003, Chicago, 12 October 2003 [online] http://www.er.byu.edu/er2003/slides/ER2003PT1Dori.pdf (accessed 17 December 2017).

Henrich, G.L., Bertelsen, S., Koskela, L., Kraemer, K., Rooke, J. and Owen, R. (2007) 'Construction physics: understanding the flows in a construction process', *15th Annual Conference of the International Group for Lean Construction*, Michigan State University, East Lansing [online] http://usir.salford.ac.uk/29157/ (accessed 15 November 2017).

ISO (2010) *ISO DIS 10628-2: Diagrams for the Chemical and Petrochemical Industry*, 15 July.

ISO/PAS (2015) *ISO/PAS 19450:2015 – Automation Systems and Integration – Object-Process Methodology*, iso.org, December.

Jackson, M. (1995) *Software Requirements & Specifications: A Lexicon of Practices, Principles, and Prejudices*, Addison-Wesley, New York, NU, USA.

Kolmetz, K., Jaya, A. and Ristiyanti, R. (2015) *Kolmetz Handbook of Process Equipment Design: Process*, KLM Technology Group [online] http://kolmetz.com/PEDG.html (accessed 10 January 20108),

Liu, H. and Gluch, D.P. (2004) 'Conceptual modeling with the object-process methodology in software architecture', *Journal of Computing Sciences in Colleges*, Vol. 19, No. 3, pp.10–21.

Luckham, D. and Schulte, R. (2011) *Event Processing Glossary Version 2.0* [online] http://www.complexevents.com/2011/08/23/event-processing-glossary-version-2-0/ (accessed 10 January 2018).

Ministry of Electricity and Water (MEW) (2017) [online] http://www.mew.gov.kw/en/?com=content&id=73&act=view (accessed 15 June 2017).

Pinkerton, N., Schmidt, K. and Xamplas, J. (2014) *Block Flow Diagram, Process Design Project*, updated 1 March 2015 [online] https://processdesign.mccormick.northwestern.edu/index.php/Block_Flow_Diagram (accessed 10 December 2017).

Reinhartz-Berger, I. and Dori, D. (2004) 'Object-process methodology (OPM) vs. UML: a code generation perspective', *CAiSE Workshops (1)*, pp.275–286, Faculty of Computer Science and Information Technology, Riga Technical University, Riga, Latvia.

Rinne, M., Blomqvist, E., Keskisärkkä, R. and Nuutila, E. (2013) 'Event processing in RDF', *Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns*, Sydney, Australia, Vol. 1188, pp.52–64.

Siddiqui, N.A., Minde, N., Mondal, P. and Vijayakumar, A.K. (2015) 'Study of P&ID, safeguarding philosophy, design basis and its features', *International Journal on Occupational Health & Safety, Fire & Environment,* Vol. 3, No. 1, pp.1–6.

SMC (2013) *Systems Engineering Primer & Handbook: Concepts, Processes and Techniques*, 4th ed., Vol. 1, Space & Missile Systems Center, U.S. Air Force [online] http://www.everyspec.com (accessed 10 January 2018).

Tuffley, D. and Wen, L. (2012) 'Behavior engineering for the analysis and verification of processes', *Advanced Materials Research*, Vols. 399–401, pp.1798–1801.

Turton, R., Bailie, R.S., Whiting, W.B. and Shaeiwitz, J.A. (2009) *Analysis, Synthesis, and Design of Chemical Processes*, 3rd ed., Prentice Hall, Upper Saddle River, New Jersey.

Turton, R., Bailie, R.S., Whiting, W.B., Shaeiwitz, J.A. and Bhattacharyya, D. (2012) *Analysis, Synthesis, and Design of Chemical Processes*, 4th ed., Prentice Hall, Upper Saddle River, New Jersey.

Vermaas, P.E. and Dorst, K. (2007) 'On the conceptual framework of John Gero's FBS-model and the prescriptive aims of design methodology', *Design Studies*, Vol. 28, No. 2, pp.133–157.