# Thinking in Terms of Flow in Design of Software Systems

Sabah Al-Fedaghi[1,a]

[1]*Computer Engineering Department, Kuwait University, Kuwait*

**Abstract.** This paper examines conceptual models and their application to the task of software system design. With the software development life cycle in mind, the concern here is the initial phases of the design process, when customer needs are investigated, conceptualized, and included in specifications. More specifically, the paper concentrates on the "patterning aspect of cognition" (i.e., object-oriented) where pattern refers to recurring templates used by designers in thinking. It is proposed that representation of thinking activity be based on flows of things using flow machines formed by stages (states) occurring sequentially in a flow. According to such an approach, a designer's "thought machine" forms a train of thought that excludes other modes such as procedural and object-oriented modes of thinking. The new idea presented here is that flow-based modelling is used not only as an external representation of the design's thinking but also as the style of thinking. A thinking style involves how one organizes thoughts and is a "conscious system of design." The method is illustrated by remodelling examples from the object-oriented paradigm. It seems to have merits that deserve further development.

## 1 Introduction

Among the many functions of the human brain is the ability to think [1]. The cognitive faculty of thinking [2] involves the processes by which we reason and solve problems. In the study of cognition, one aspect is consideration of conceptual models of how we view the world [3]. This paper examines conceptual models and their application to software system design. With the software development life cycle in mind, the concern here is the initial phases of the design process, when customer needs are investigated, conceptualized, and included in specifications. More specifically, the paper concentrates on the "patterning aspect of cognition" [4] (e.g., object-oriented) where pattern refers to recurring templates used by designers in the thinking process.

In general, cognitive models are characterized by great variation depending on factors that include society, environment, and culture (e.g., science, religion) that reflect diverse perceptions of the way the world works. Persons may hold onto more than one of these models and, depending on context, *adopt* a particular view of the world as a framework for thinking. Such selected model plays an important role in thinking by "automatically" constraining meanings (again, e.g., object-oriented). It is like wearing glasses that affect what we see and cannot see. The models adopt a certain way of generating their elements (e.g., space in architectural models), and they direct attention in interactions with the world and affect how we link concepts to objects and vice versa.

[a] sabah.alfedaghi@ku.edu.kw

A model of such type could be founded *predominantly* upon certain types of mental components such as feelings, concepts (ideas), propositions, and pictures. A *conceptual* model is a mental model formed mainly upon concepts as components of thinking. A model can provide a framework for thinking that structures concepts into patterns according to categories. A model can also provide a basis to represent internal thinking in an external form.

Diagrams, as one form of representation, convey a simple and easily communicable method of thinking. "Diagrams are essential representations for thinking, problem solving, and communication in the design disciplines" [5]. Because "Diagrams are effective instruments of thought" [6], this paper suggests imposing a specific style of thought such that (internal) thinking and (external) representation use the same model, thus synchronizing thinking with the method of diagramming. According to such an approach, a designer's "thought machine" forms a train of thought that excludes other modes such as procedural and object-oriented modes of thinking.

The next section introduces a flow-based diagrammatic language to be used both as a thinking style and as a vehicle for drawing the outcome of a design. Section 3 applies such an idea to samples taken from the object-oriented literature.

## 2 Diagrammatic language for thinking and drawing

This section includes two subsections. The first will review a flow-based model (this example is a new contribution) and the second will use such a model to explain the aim of the research in this paper.

### 2.1 Diagrammatic Language to be used in modelling

We propose that thinking and drawing activities be based on *flows of things* grounded in what we call the Flowthing model (FM). FM has been used in a variety of applications (e.g., [7-12]). FM depicts processes using *flow machines* formed by five stages: creation, processing, release, transfer, and receive, as shown in Fig. 1. Hereafter, flow machines will be referred to as machines.

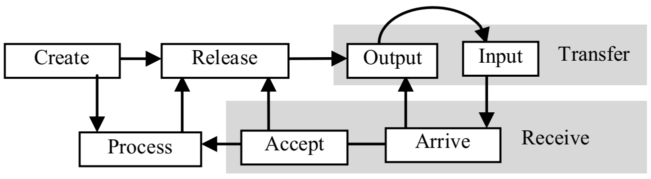

**Figure 1**. Flow machine

FM is a method for modeling things that "flow"; i.e., things that are *created*, *processed*, *released*, *transferred*, and *received* while retaining their individuality throughout the flow. They flow within *spheres*, i.e., the relevant environment that encompasses the flow. Things that flow are referred to as *flowthings*, or simply as *things*. *Flow* in such a model conveys (conceptual) movement in a system. The life cycle of a thing is defined in terms of the five mutually exclusive *stages* of creation, release, transfer, arrival, acceptance, and process (in this last stage the *form of a thing may be changed*, but no new thing is generated). For example, a *flow thing* that is *received* is not permitted to move (flow) to the stage in which it is *created* (i.e., a received thing implies an already existing thing, hence it is not possible to re-create it). A mechanism that can change this succession of flow is called *triggering*. In a propositional sense, flows of things are formulated in terms of the propositions *Create(thing)*, *Process(thing)*, *Receive(thing)*, *Release(thing)*, and *Transfer(thing)*. The reflexive arrow in Fig. 1 indicates flow to the *Transfer* stage of another machine. For simplicity, the stages Arrive and Accept can be combined and termed *Receive*.

FM is a generalization of the model input → Process → output. According to Deepak [13],

> If a system in *any field* is analyzed. It will be observed that it has three basic parts, which are organized in an orderly manner. These three parts can be represented in a model as [input → Process → output]… When a system is designed to achieve certain objectives, it automatically sets the boundaries for itself. The understanding of boundaries of the system is essential to bring clarity in explaining the system components and their arrangements.

**Example**: Peters and Langemann [14] developed a general supply chain of the human brain, illustrated by a type of bubble diagram (not shown), as follows:

> Energy from the remote environment is brought to the immediate environment, it is then taken up by the body, and from there a large part of it enters the brain. In a general supply chain, the flux of energy can principally be determined by the supplier (previous step) or by the receiver (proximate step). The share of the flux which is determined by the supplier is called the "push component" …, the share which is determined by the receiver is called the "pull component" ....

Fig. 2 shows the FM representation of a partial picture of their diagram. The arrow represents the flow of energy. For the sake of simplicity, the energy machine is not enclosed in its own box as a subsphere.

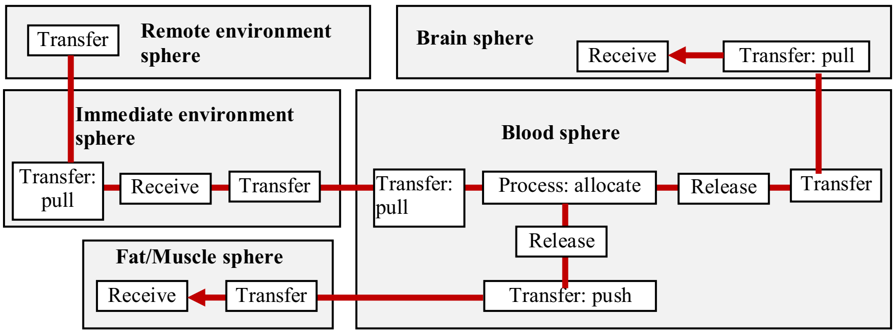

**Figure 2**. FM representation of a general supply chain of the human brain

## 2.2 Aim of This Research

In design, "Drawing is intimately bound with thinking" [5]. We propose imposing flow-based thinking on the conceptual design process instead of other types of thinking such as categories and types that trigger other types (e.g., object-oriented). In forcing this FM-based thinking, we isolate and cut nature up into *things that flow in their spheres* (environment).

Accordingly, we utilize FM in this paper as a machine that unifies the design process to achieve the following dual purposes:

1. A designer's "thought machine" that excludes other modes of thinking such as procedural and object-oriented modes of thinking (see Fig. 3). "Every person is conscious of a train of thought being immediately awakened in his imagination analogous to character or expression of the original object" (Gandy [15], quoted in Crook [16]).

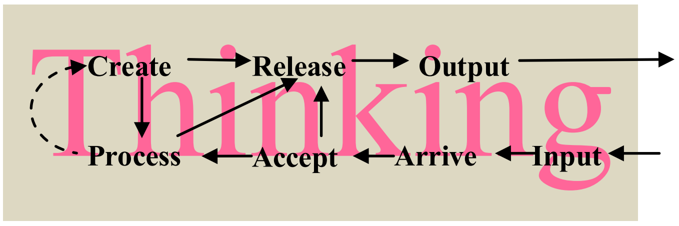

**Figure 3**. Thought machine

2. A "design outcome" in the form of a representative diagram that models a portion of reality, as shown in Fig. 4. Note that the figure is a version of the famous semiotic triangle of object, thought, and sign (representation) [17].

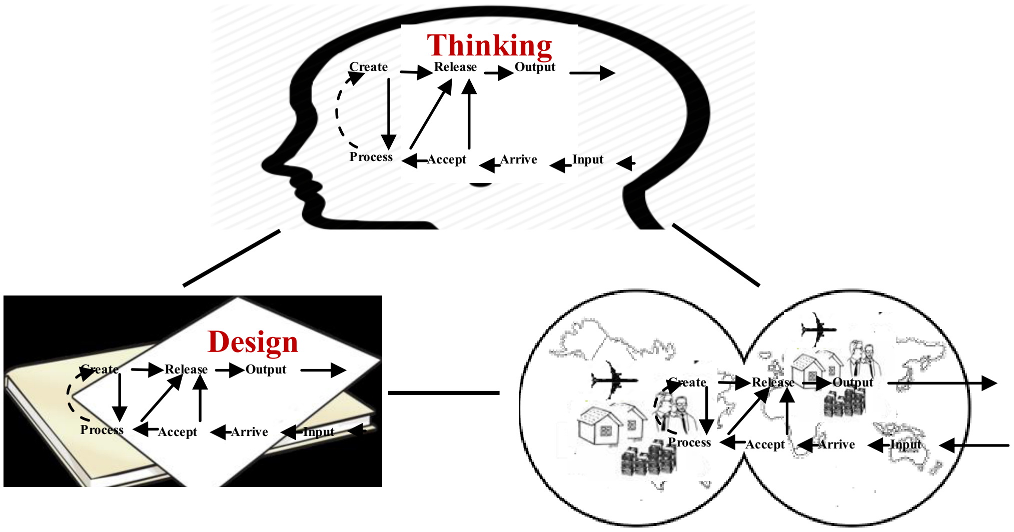

**Figure 4**. FM-based world perception, thinking and design diagramming

## 3 Thinking style

The new idea in this paper is that FM can be employed *explicitly* not only for the drawing of an external representation but also as the *style of thinking*. A thinking style involves how one organizes thoughts. In architecture, according to Crook [16], a style is a "conscious system of design." This scheme would have wide application in many areas besides software design, as follows:

•In mathematics, a *set* can be conceptualized as a flow machine that handles flowthings called elements that flow through and into and out of the system. Handling here refers to transferring, receiving, processing, creating, and releasing elements of the set. A function such as f: X → Y can be represented in terms of a map between sets, conceptualized not in terms of space and movement, but in terms of flows in which the values of input flowthings are changes in the output flowthings. Such a view may have some application in teaching of mathematics [7].

•Physical security can be modeled as a flow machine of interruptions or breaks with an abstract apparatus that represents the flow of things in the physical infrastructure of the organization [9].

•According to Gandhi, "Man is the most wonderful machine in creation." Accordingly, such notions as privacy can be conceptualized as an inner flow machine of a person. It is a natural machine whose function is to control the transferring of (personal) things to the outside [8].

Other applications of flow-based thinking include the areas of aesthetics [10], philosophy [11], and computer programming [12].

According to Penrose [18], "There seem to be many different ways in which different people think – and even in which different mathematicians think about their mathematics." Likewise, in design, a variety of methods of *thinking* apply to design. "You cannot hold a design in your hand. It is not a thing. It is a process" [19].

This paper proposes that the designer think in terms of FM when perceiving reality, in addition to drawing FM diagrams. We speculate that flow-based thinking is a thinking style. Even if this style does not seem "natural," the claim is that thinking and drawing this way lead to a viable design methodology, especially in the area of software development.

One "thought thing" follows another, connected by flows (including triggering), and the connection is driven by such principles as contiguity in time or place, and cause and effect (these principles were laid out by Hume in *An Inquiry Concerning Human Understanding*, 1748).

Accordingly, guidelines for devising an FM design for a particular problem are as follows:

- "Pick up" a thing
- Follow its flow through spheres until
- It triggers another thing that leads to a new stream of flow to follow, etc.

For example, the flow of an *order* leads to the flow of an *invoic*e, which leads to a flow of *payment*, which leads to the flow of the *ordered product*.

"Design without drawing seems inconceivable… Sketches can reveal thought, then, ... [and] they can reveal the elements or segments of construction or of thought in a particular domain" [20]. Early in the design process, designers draw diagrams as a way to map their thinking and explore concepts (mental things) and their flows in an iterative thinking process (thinking$_1$ → FM$_1$, thinking$_2$ → FM$_2$, …; see Fig. 4). This process involves capturing ideas and their flow in a succession of conceptual phenomena. Drawing is a vehicle for seeing ideas and their flows and triggering, hence for understanding and organizing their system.

In the context of FM, we say that FM can provide a basin for the flow of thought when the designer thinks in terms of FM, and that "elements and segmentation construction of thought" appear as elements of flow: the conceptual movement of things along the stages of the flow systems. The first phase of design includes identifying end-user motivations and needs, which form the basis of thinking that generates concepts to meet these needs. Arbitrary types of "thinking happenings" drive the appearance of concepts in disconnected thoughts with lack of clear connections between thoughts, just as butterflies, weighing no more than feathers, are blown about in the wind. This paper formulates this phenomenon ("thinking happenings") in a restrained way such that the thinking gives birth and directs *ideas (concepts)* like a controlled wind flow that blows butterflies into a particular air tunnel.

The main notion is that adopting an FM-based *conceptualization* (cognition) precedes its instantiation as a diagram, just as object-oriented thinking leads to object-oriented languages. This idea is analogous to the mapping between abstract classes in the mental domain and objects and their classes as data structures in object-oriented models. The mapping between FM-based thought and an FM-based diagram ought to be possible when the same conceptual pattern underlies each.

With FM, a small number of notions (flow, stages, spheres) with very little verbal description are used to construct a large conceptual picture and physical FM-based depiction. At one level, things can be viewed as concepts, and flows as the conceptual movement of concepts in their spheres. On the other hand, a diagram can be viewed as a composition of arrows and boxes that reflects the conceptual picture through motor elements involved in the act of drawing, i.e., physiological signals and physical actions.

In contrast, the object-oriented (OO) approach focuses on objects that have attributes and do processes; e.g., in GUIs, windows and icons have attributes (locations on the screen), and they can be clicked or moved [13]. In this case, particular "classes" of objects are defined. The class definition is the master description of what data the instances will have and what can be done with that data. We make new instances based on classes [13]. Deepak [13] provides a set of guidelines for formulating an OO design for a particular problem:

- Describe the problem in plain English. Describe what the types of entities in the problem are, what attributes they possess, what actions they can take, and what those actions need to work.
- Write a class for each type of entity, where the attributes are instance variables in the class, and the actions are its methods, and what the actions need to work are passed as parameters to those methods.
- Instantiate instances of the objects you need, and call their methods [13].

## 4 Sample styles of thinking about problems

This section provides some evidence for the viability of FM as a foundation for thinking and drawing through modelling examples from the object-oriented paradigm.

### 4.1 Example 1

Manik [21] gives an example of an object-oriented model, shown in Fig. 5. The *lives in* and *drives* associations, in the OO approach, indicate instances of classes associated with each other. We can add attributes to the association with an association class (connected to an association by a dotted line, not shown). It seems that the model was developed using Deepak's [13] guidelines for OO design reviewed above.

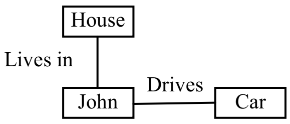

**Figure 5**. An object-oriented model (partial, re-oriented from [21])

According to FM and within the context of the problem under consideration, we first "pick up" a flowthing, say*, John* (Fig. 6a) to " jumpstart" our train of thought. This beginning is complemented by recognizing John's spheres (Fig, 6b and c), then identifying the flow of John; consequently, this step leads us to model the car as shown in Fig 6d. The OO modeling involves conceptual discontinuity that jumps from John to house and to car, resulting in disconnected entities. The designer must examine these disparate concepts pair-wise to jump between John/house and John/car and their relationship. An alternative scenario involves somehow recognizing (e.g., by reading the whole English text – see guidelines in [13]), that *John* is the central entity and can be used as a reference point, then looking around (e.g., reading the text again) to discover the *house,* connect it to John, then discover the *car* and make the second connection. While such an approach is accepted as standard by many designers, this description emphasizes the difference from the FM thinking approach.

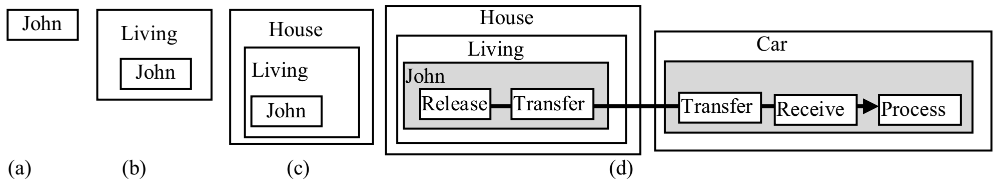

**Figure 6**. Stages in flowthing modelling.

FM modelling is a smooth maneuver along a flow, analogous to holding onto a rope (the flow) and walking along it to reach the next thing to be modeled; e.g., John leaves the *house*, reaches the *car*, starts it up, and drives it away.

Note that Fig. 6c is the "natural" conceptualization of *John lives in a house*. It brings to mind a person *inside* his house, as shown in Fig. 7 (left), with stick figure inside an FM sphere of House. Even searching Google for "lives in + house" results in a similar illustration of a man sitting in a house (Fig. 7 (right)). However, OO modelling reflects a very abstracted portrait of reality developed for the sake of data structuring. If this is the aim of modelling, then to reach such modelling based on data structure, we can:

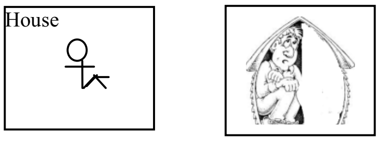

**Figure 7**. FM conceptual representation of "John lives in a house" and image shown in a Google hit when the term ""lives in + house" is searched

- Remove the flow and separate each sphere as in Fig. 6d, resulting in Fig. 8, and
- Replace living and process by lines in that figure, so we get the original OO representation of Fig. 5.

Accordingly, FM representation can be used as a semantic base for OO modelling.

We speculate that FM is closer to ordinary thinking. In fact, the FM representation "pulls" us to follow our thinking to its end when conceptualizing *John lives in a house and drives a car*, as shown in Fig. 9. The figure expresses a portrait of the world of John: John lives in a house, leaves the house to drive a car, drives the car back to the house, gets out of the car, and enters the house.

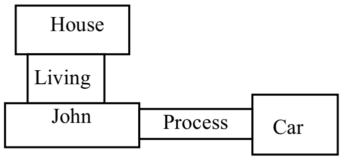

**Figure 8**. Abstracting the FM representation in Fig. 6d.

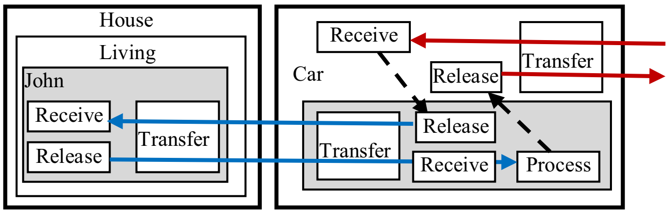

**Figure 9**. Complete description of Jon lives in a house and drives a car.

The result is a complete representation; e.g., when designing a house on this structure, its bounds and a door (representing transfer) immediately come to mind. Similarly, *John lives in a house and John drives a car* brings to mind that John is the common component in this limited world encompassing a house and a car; hence John moves (flows) from the house to the car and vice versa. This is represented in the OO model by having the two lines connected to John.

Additionally, the OO model embeds ambiguity by using lines that represent associations. *Drive a car* has two types of "relationships" with *lives in a house*, in addition to being related to John:

- The car and the house form a "house transportation system" of going away and coming back to the house by use of the car.
- *Lives in a house* and *Drives a car* are separate spheres of *house living* and *driving a car* (to somewhere else).

For example, in the relational database model, the first relationship is represented by a three-tuple real table while the second one is represented by two tables. In the last case, building a universal table may violate the multi-value dependence principle if John lives in more than one house and drives more than one car.

The given FM representation in Fig. 9 assumes the first case, and Fig. 10 shows the second case. Note that *Create* in the sphere of John denotes "come into existence" and is omitted in the spheres of House and Car for simplicity, since specifying the flow indicates their existence. The OO model could be interpreted to be either of these two cases.

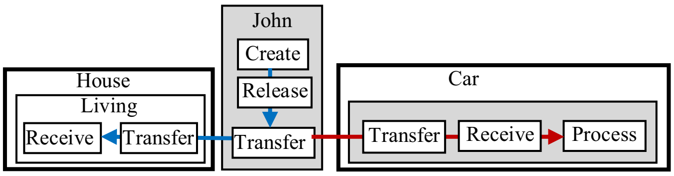

**Figure 10**. *Lives in a house* and *Drive a car* are separate spheres

## 4.2 Example 2

In his book *Introduction to Object Oriented Programming*, Budd [22] illustrates the concepts of OO with an example of *sending flowers to a friend in a different city*, as follows.

- Chris is sending flowers to Robin.
- Chris can't deliver them directly. So Chris uses the services of the local Florist.
- Chris tells the Florist (named Fred) the address for Robin, how much to spend, and the type of flowers to send.
- Fred contacts a florist in Robin's city, who arranges the flowers, then contacts a driver, who delivers the flowers (description taken from [23])

The OO approach handles this "real-world situation" by representing it as shown in Fig. 11. "The Object Oriented approach is to view software the same way we view the world – easier for us to understand… The chain reaction that ultimately resulted in the solution to Chris's problem began with a request given to the florist Fred. This request led to other requests, which led to still more requests, until the flowers ultimately reached Chris's friend Robin." [23].

Ferens [24] developed a UML instance diagram for the same problem, shown in Fig. 12. Note the abstracted representation that ignores the differences among *things* in the figure where Chris sends the *Order* and the Driver carries *physical flowers*. Also, the structure is ignored, e.g., the driver is within the sphere of Robin's florist, not any driver.

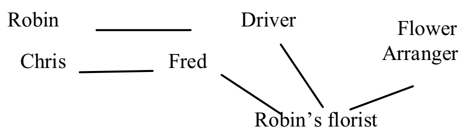

**Figure 11**. Representation of sending flowers to a friend in OO modeling (partial, redrawn from [23])

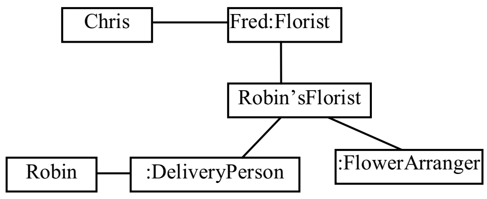

**Figure12**. UML Instance Diagram (partial, redrawn from [24])

According to Weiss [23], the chain reaction that results in the solution to the problem of *sending flowers to a friend* starts with a request to florist Fred that leads to other requests until the flowers reach Robin. This chain is based on the transmission of a message to an agent/object. A message is either a request for information or a request to perform an action.

Fig. 13 shows the FM representation of the sequence of activities involved in sending flowers to a friend.

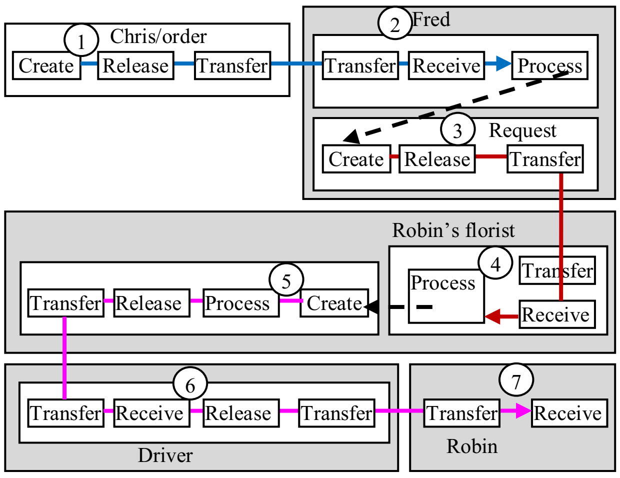

**Figure 13**. FM representation of *sending flowers to a friend*

We assume that the modeler is *thinking of a* starting *flowthing* to follow the *thread of flow* and "pull up" the rest of the flows. Accordingly, the first thing that comes to mind is the first item, *Chris*, and the flow is *order*ing of flowers.

- The *order* (circle 1) flows to Fred (2), who *processes* it. Note that, for simplicity, we depict Chris and order in one box.
- This processing triggers creation of a request (3) that flows to Robin's florist (4).
- Receipt of the request triggers creation of the flower basket per instructions in the order (5).
- The basket flows to the driver (6).
- The basket is delivered to Robin (7).

## 5 Conclusions

This paper has proposed basing the thinking behind system design on flows of things passing through flow machines. According to such an approach, a designer's "thought machine" forms a train of thought that excludes other modes such as procedural and object-oriented modes of thinking. The paper emphasizes this thinking style as a unifying method that could have diverse applications. The re-modelling of examples from the object-oriented paradigm seems to point to merits that deserve further development.

## References

1. T. Parry and G. Gregory, (1998). Designing Brain Compatible Learning. Victoria: Hawker Brownlow Education, pp.168-206 (1998).
2. R. Langacker, Foundations of Cognitive Grammar, vol. 1: Theoretical Prerequisites. Palo Alto: Stanford University Press (1987).
3. M. Wagman, Cognitive Science and Concepts of the Mind: Towards a General Theory of Human & Artificial Intelligence. New York: Praeger (1991).
4. R. C. Anderson, "The notion of schemata and educational enterprise: general discussion of the conference," in R. C. Anderson, R. J. Spiro, and W. E. Montague, Eds., Schooling and the Acquisition of Knowledge. Hillsdale, N.J.: Erlbaum (1977).
5. E. Y.-L. Do and M. D. Gross, "Thinking with diagrams in architectural design," Artif. Intell. Rev., vol. 15, pp. 135-149 (2001).
6. N. Crilly, A. F. Blackwell, and P. J. Clarkson, "Graphic elicitation: using research diagrams as interview stimuli," Qual. Res., vol. 6 (2006).
7. S. Al-Fedaghi, "Thing-oriented learning: application to mathematical objects," 19th IEEE International Conference on Computational Science and Engineering (CSE 2016), Paris (2016).
8. S. Al-Fedaghi, "Privacy as a machine," International Conference on Information System and Artificial Intelligence (ISAI2016), Hong Kong (2016).
9. S. Al-Fedaghi, "Security as a flow machine," IEEE World Congress on Industrial Control Systems Security (WCICSS-2015), London (2015).
10. S. Al-Fedaghi, "Schematization for machine-oriented aesthetics," 15th International Conference on Artificial Intellegence, Las Vegas, USA (2016).
11. S. Al-Fedaghi, "An engineering reading of Deleuze and Guattari's Anti-Oedipus," 2016 International Symposium on Economics and Social Science (ISESS 2016), Bangkok, 2016.
12. S. Al-Fedaghi, "Conceptual understanding of computer program execution: application to C++", Int. J. Comp. Sci. Iss. arXiv, vol. 10, no. 2, March (2013).
13. K. Deepak, Management Information System [blog posting], June 9 (2011). http://deepread.blogspot.com/2011/06/concept-of-system-analysis-design-sad.html
14. A. Peters and D. Langemann, "Build-ups in the supply chain of the brain: on the neuroenergetic cause of obesity and type 2 diabetes mellitus," Front. Neuroenergetics, vol. 1, no. 2, p. 2, February (2009).
15. J. Gandy, "Architecture, its natural model,".Exhibited at the R.A., Soane's Museum, SM XP12(1838) .

16. J. M. Crook, The Dilemma of Style: Architectural. Ideas from the Picturesque to the Post-Modern. Chicago: University of Chicago Press (1987).
17. C. K. Ogden and I. A. Richards, The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. London: Routledge & Kegan Paul (1923).
18. R. Penrose, The Emperor's New Mind. Vintage, London (1990).
19. B. Gill, Graphic Design as a Second Language. Images Publishing (2004).
20. B. Tversky, “What does drawing reveal about thinking?” in J. S. Gero & B. Tversky, Eds., Visual and Spatial Reasoning in Design. Sydney, Australia: Key Centre of Design Computing and Cognition., pp. 93-101 (1999).
21. S. M. Manik, Object Oriented Concept [slides], Feb 29, 2012. http://www.slideshare.net/smj/object-oriented-concept
22. T. Budd, An Introduction to Object-Oriented Programming. Addison Wesley Longman, (2002). ISBN 0-201-76031-2
23. G. Weiss, “Slides based on Timothy Budds‘s slides” [course materials], (No date). https://www.google.com.kw/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwjb-a37hL3PAhXJAsAKHYbDCUYQFgggMAE&url=https%3A%2F%2Fwww.cs.bgu.ac.il%2F~oosd112%2Fwiki.files%2Fl1.pdf&usg=AFQjCNEms3HJzDe6wOK8mUgUrWrw9v5x5A&bvm=bv.134495766,d.bGg
24. K. Ferens, “Fundamental concepts (principles) of object oriented programming,” ECE 3740 lecture notes, Fall (2016).