Informing Science:
the International Journal of
an Emerging Transdiscipline

*An Official Publication
of the Informing Science Institute
InformingScience.org*

Inform.nu

Volume 20, 2017

FLOW-BASED PROVENANCE

Sabah Al-Fedaghi
Computer Engineering Department,
Kuwait University, Kuwait
sabah.alfedaghi@ku.edu.kw

ABSTRACT

Aim/Purpose — With information almost effortlessly created and spontaneously available, current progress in Information and Communication Technology (ICT) has led to the complication that information must be scrutinized for trustworthiness and provenance. Information systems must become provenance-aware to be satisfactory in accountability, reproducibility, and trustworthiness of data.

Background — Multiple models for abstract representation of provenance have been proposed to describe entities, people, and activities involved in producing a piece of data, including the Open Provenance Model (OPM) and the World Wide Web Consortium. These models lack certain concepts necessary for specifying workflows and encoding the provenance of data products used and generated.

Methodology — Without loss of generality, the focus of this paper is on OPM depiction of provenance in terms of a directed graph. We have redrawn several case studies in the framework of our proposed model in order to compare and evaluate it against OPM for representing these cases.

Contribution — This paper offers an alternative flow-based diagrammatic language that can form a foundation for modeling of provenance. The model described here provides an (abstract) machine-like representation of provenance.

Findings — The results suggest a viable alternative in the area of diagrammatic representation for provenance applications.

Future Research — Future work will seek to achieve more accurate comparisons with current models in the field.

Keywords — conceptual representation, provenance, diagrammatic representation, workflow, data provenance

Accepting Editor: Kay Fielden │ Received: November 14, 2016 │ Revised: February 8, 2017 │ Accepted: February 10

Cite as: Cite as: Al-Fedaghi, S.. (2017). Flow-based provenance. *Informing Science: the International Journal of an Emerging Transdiscipline, 20,* 19-36. Retrieved from http://www.informingscience.org/Publications/3665

INTRODUCTION

The *general* goal of this paper is to expand understanding of the cross-cutting issue of management of data provenance – the origin, context, and history of data – through an exploration of provenance modeling. Provenance is an overloaded term with multiple definitions (Ram and Liu, 2009), and this necessitates narrowing the context of inquiry by focusing on the notion of *data* (hence, *information*) provenance as an *informing* "vehicle" in the communication process. Accordingly, the next subsection puts provenance in the framework of informing science, followed by subsections reviewing the term provenance itself and describing the research problem addressed in the paper: modeling of provenance.

PROVENANCE IN THE CONTEXT OF INFORMING SCIENCE

Informing science is "the field of inquiry that attempts to provide a client with *information* in a form, format, and schedule that maximizes its *effectiveness*" (Cohen, 1999 [italics added]). The *informing system model,* a cornerstone of informing science, is based essentially on the model developed by Shannon and Weaver (1949) for the purpose of explaining and engineering transfer of telephone signals (Cohen, 2009; Gill & Bhattacherjee, 2009). According to Travica (2014), informing science scholars have made certain adjustments to the informing model, as depicted in Figure 1, where the receiver is renamed "client," in agreement with the assumed purpose of informing science to cater to the receiver's needs; that is, to "provide the clientele with information in a form, format, and schedule that maximizes its effectiveness."

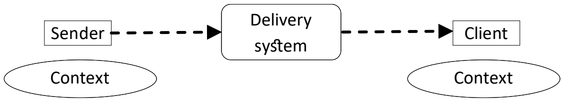


Figure 1. The model of informing science (redrawn from Travica (2014)

Such a model lacks Cohen's (1999) original element of *effectiveness*. Effectiveness denotes *informing-ness*, where the delivery of information is accompanied by *actual (effective) change* in the *informational* state of the client. *Informing* is necessary for various purposes such as understanding, learning, and decision making. A communication process may involve *informing* information or *non-informing* information (e.g., entropy equals one – no surprise) with respect to the client, measured in terms of *effectiveness*. Informing-ness depends on several factors, including trustworthiness, accuracy, truth, and the *provenance* of information. Accordingly, an informing system can be modeled as shown in Figure 2, where the emphasis is on provenance. The figure, with language to be explained a little later in this paper, expresses the notion that information with its provenance enhances informing-ness.

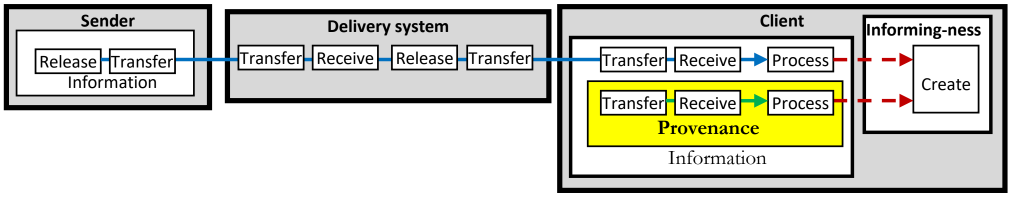


Figure 2. A model of Informing System (See the diagrammatic language proposed later)

PROVENANCE

The importance of assessing informing-ness has become increasingly important in current ICT progress as information is almost effortlessly created and spontaneously available, resulting in the vital necessity of scrutinizing its trustworthiness and provenance.

Provenance, or meta-information about the origin, history, or derivation of an object, is now recognized as a central challenge in establishing trust and providing security in computer systems, particularly on the Web. The lack of adequate provenance information can cause (and has caused) major problems, [that] can arise either from failure to disclose some key provenance information to user, or from failure to obfuscate some sensitive provenance information (Acar, Ahmed, Cheney, & Perera, 2013).

Provenance is one of the core aspects influencing the trustworthiness of information (Nurse et al., 2011), where a main issue is: what was the process that led to the information in question? The term *provenance* is typically used to refer to the trusted, proven history of certain data to achieve authority and importance. The World Wide Web Consortium (W3C) Provenance Working Group defines provenance as "information about entities, activities and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness" (W3C, 2013). The focus in this context is on handling the metadata related to the origin, chain of custody, and derivative ancestry or process that yielded the data in question (Cheney, 2010; Nassopoulos, Serrano-Alvarado, Molli, & Desmontils, 2015). There is a temporal relationship between original data versions and newly created data as a side effect of operations applied to these data.

Historically, discovering and making sense of archival data is a basic intellectual task (Stanford Encyclopedia of Philosophy, 2012). Currently, provenance is a critical issue since use of computers and networks has stimulated the formation of huge amounts of data about the past that can be used to answer questions concerning past creation, processing, and transmission of a particular piece of data.

Science, industry, and society are being revolutionized by radical new capabilities for information sharing, distributed computation, and collaboration offered by the World Wide Web. This revolution promises dramatic benefits but also poses serious risks due to the fluid nature of digital information. One important cross-cutting issue is managing and recording provenance, or metadata about the origin, context, or history of data. We posit that provenance will play a central role in emerging advanced digital infrastructures (Cheney, Chong, Foster, Seltzer, & Vansummeren, 2009).

Provenance is crucial for determining the credibility of data. "If you are a scientist, or any kind of scholar, you would like to have confidence in the accuracy and timeliness of the data that you are working with. In particular, you would like to know how it got there" (Cheney, Chiticariu, & Tan, 2007). According to Acar et al. (2013), "Many computer systems will need to become provenance-aware in order to provide satisfactory accountability, reproducibility, and trust for scientific or other high-value data." The difficulty is that information and communication technologies have made it easy to copy and transform a wide variety of data types.

Provenance applied to data is concerned primarily with the history of a particular piece of data (Cheney et al., 2009). The common forms of data provenance include tracking the "sources" of and reasons for data (Nassopoulos et al., 2015; Buneman, Khanna, & Tan, 2000), describing how an output record was produced (Green, Karvounarakis, & Tannen, 2007; Green et al., 2007), and knowing the source data that produced the specified data (Cui, Widom, & Wiener, 2000).

Research Problem: Provenance Modeling

Several models have been proposed for recording provenance of data (e.g., Agrawal et al., 2006; Green et al., 2007a, 2007b; Ikeda & Widom, 2010). Research in this area encompasses work on representation models, as well as on procedures and agents that create and process data. Developing standard models for capturing and publishing provenance of artifacts resulted in development of the Open Provenance Model (OPM) (Moreau et al., 2011) for depicting provenance graphs in terms of a workflow system. More recently, the W3C developed a data model for provenance (PROV) that describes the entities, people, and activities involved in producing a piece of data or thing (W3C, 2016), but according to Cuevas-Vicenttín et. al. (2016),

[These models] do not suffice for encoding scientific workflow provenance. The reason being, that both OPM and PROV were developed as minimal models meant to be used for tracking the provenance of resources on the Web regardless of their types. As such, they do not provide all the concepts that are necessary for specifying workflows and encoding the provenance of data products used and generated... Thus, the need arises for a new model.

A generally followed approach to provenance involves the general model of input–process–output, where additional metadata are recorded to provide a description of how the output was obtained (Cheney, 2010). According to Nassopoulos et al. (2015), the two main categories of provenance approaches are workflow provenance and data provenance. Workflow provenance aims to capture a description of the workflow of the involved system that specifies transactions and their interactions (Nassopoulos et al., 2015). According to Cheney (2010), workflow provenance semantics are usually specified informally, resulting in a confusing variety of models and styles of provenance for workflows.

Instances of [the Open Provenance Model] are graphs whose nodes represent agents, processes or artifacts and whose edges represent dependence, generation or control relationships. The OPM has "semantics" in the sense of the Semantic Web, in that the nodes and edges are expected to have names that are meaningful to reasonably well-informed users. (Cheney, 2010)

This paper is a contribution to this area with a proposed flow-based diagrammatic model that can form a foundation for provenance. Without loss of generality, the focus here is on depiction of an OPM description of provenance in terms of a directed graph. Several case studies of OPM-based representation are taken as benchmarks to examine how well this modeling method represents these cases in comparison with OPM.

Diagrammatic notations facilitate "understanding and promote a shared visual representation" (Moreau et al., 2011). According to Archer (2011), "Provenance relationships among data are *naturally* represented in graphs" [italics added]. Specifically, at this stage of development of the proposed model, we utilize *human understanding of diagrams* as a tool for provenance studies by contrasting an OPM graph with the representation proposed in this paper.

According to Ram and Liu (2009), due to the lack of consensus on the semantics or meaning of provenance, current efforts on capturing data provenance have focused on only one or two aspects of provenance while ignoring others. As a result, the provenance is often incomplete and cannot be shared across applications. In response to this challenge, we attempt to formally define the semantics of provenance that can be agreed upon by people from different domains.

Our proposed model captures provenance semantics in a flow-based diagram that depicts different stages of flow in the life cycle of a "thing."

Summary of Contents

As background for our proposed approach to provenance and for the sake of a self-contained paper, the review in the next section summarizes the general features of a model that has been adapted for varied applications (Al-Fedaghi, 2016a-e, 2017a, b). The OPM example in the section is a new contribution.

To illustrate the problem of diagrammatic representation of provenance and the relevance of our contribution, the section titled APPLICATION TO DATABASE TABLES discusses a specific example given by Müller (2016) in the context of provenance using SQL. According to Müller (2016), relational databases have a deficiency in provenance computation for SQL.

That section is followed by two main sections, PROVENANCE OF A CAKE and PROVENANCE OF A COFFEE SHOP, in which the proposed modeling language is applied to two known modeling case studies from the literature of provenance.

Diagrammatic Language: Review

This section summarizes the Flowthing Machine (FM) model, which provides a diagrammatic language proposed as a high-level description suitable for provenance-based applications. FM is a generalization of the well-known input–process–output model, utilized in many scientific fields. It involves handling of *flow things*: things that can be *created*, *processed*, *released*, and *transferred*, *arrive,* and *be accepted* as shown in the *flow system* depicted in Figure 3. If all things that arrive are accepted, the *Arrive* and *Accept* stages can be combined in a single *Receive* stage. Hereafter, a *flow thing* will be referred to merely as a *thing*. The arrows in the figure represent flows of things

The environment of flow is called its *sphere*, e.g., data flow within the sphere of a company. As an example, a simple communication system can be represented as shown in Figure 4. The sender and receiver are each spheres, and the message is a *thing* that flows in the message *flow system*. Note that a flow system itself is a special type of sphere.

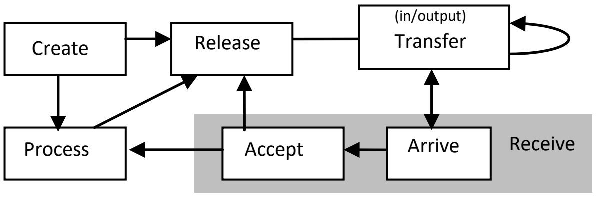


Figure 3. Flow system.

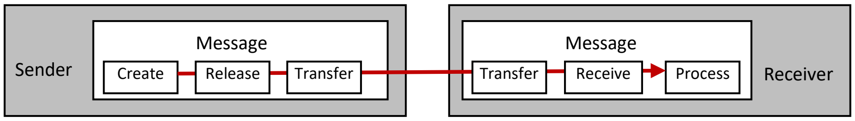


Figure 4. Example of simple communication.

The stages of a flow system are mutually exclusive; that is, a thing always exists in one and only one of these states or stages at any moment. Assuming the thing is a datum, *Process* in this model is any operation on the datum that does not produce a new piece of data. *Creation* denotes the appearance of a new datum in the flow system.

There are many types of *things*, including data, information, money, food, fuel, electrical current, and so forth. The life cycle of a thing is a sequence of stages through which it passes in a stream of flow. Other “states” of things, for example, *stored*, are secondary states; thus, we can have a *stored created* thing, a *stored processed thing*, and so forth.

In addition to flows denoted as arrows, FM includes triggering mechanisms represented by dashed arrows. Triggering denotes activation, such as starting a new flow, as exemplified in Figure 5.

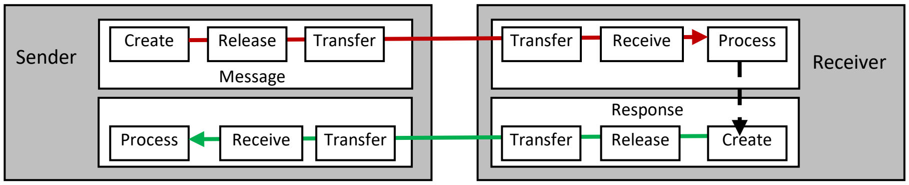


Figure 5. Example of simple communication that triggers a response.

Example: A primary concern of OPM is being able to represent “things.”

It is recognized that many of such “things” can be *stateful*: a car may be at various locations, it can contain different passengers, and it can have a tank full or empty; ... Hence, from the perspective of provenance, we introduce the concept of an *artifact* as an immutable piece of state… which may have a physical embodiment in a physical object, or a digital representation in a computer system. (Moreau et al., 2011).

In FM, a car is a *sphere* that includes the physical car and its properties, as shown in Figure 6. When the car *itself* flows it is understood that its properties flow with it, until some point at which it is necessary to present the properties explicitly. Thus, in place 1 in Figure 6, the sphere of the car is declared and can be initialized, e.g., 5 passengers. The car (as a physical thing) flows to place 2 with no change in its description. When it arrives at place 3, there is a change in the number of passengers, thus it is necessary to show the number of passengers. The point here is that the car can contain several flow things. If the description of concern in the model is just the physical car (place 2), where its attributes do not change, then the assumption is that these attributes flow with the car. An observer at place 1 sees the car and its attributes, whereas in place 2 a second observer sees just the physical car passing by.

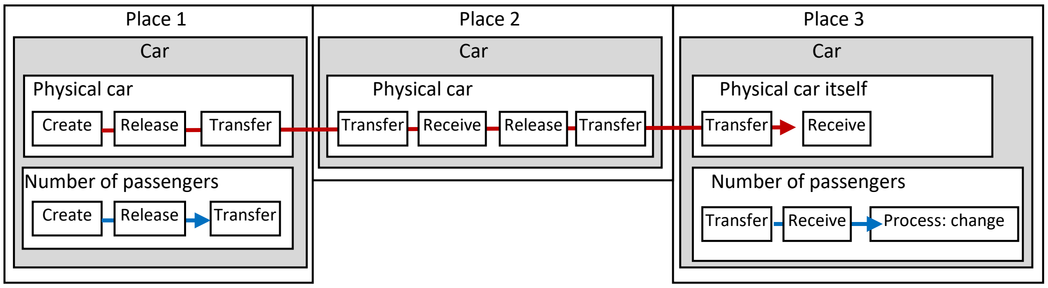
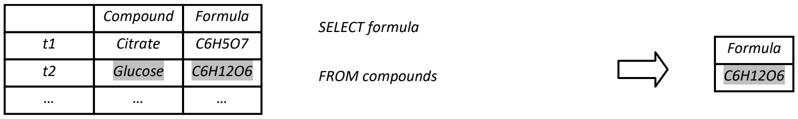


Figure 6. FM representation of a car as described in OPM.

Application to Database Tables

To illustrate the problem of diagrammatic representation considered here and our proposed solution, this section discusses a specific example given by Müller (2016) in the context of provenance using SQL. According to Müller (2016), relational databases have a deficiency in provenance computation for SQL, as shown in Figure 7. In the figure, the data provenance of the process that results in "Formula C6H12O6" is indicated by the shaded tuple in the table.

> According to the SQL query, two input columns are accessed: *Compound* is used to decide if a tuple gets filtered or not. If a tuple qualifies, its value sitting in *Formula* is copied over into the result table. Our provenance analysis accordingly finds the result being why-dependent on tuple t2: *Glucose* and being where-dependent on t2: *C6H12O6*. (Müller, 2016)



	Compound	*Formula*
t1	*Citrate*	*C6H5O7*
t2	*Glucose*	*C6H12O6*
...	...	...

SELECT formula

FROM compounds

⇨

Formula
C6H12O6

Figure 7. A table and query (redrawn, partial from (Müller, 2016)).

This explanation answers the question, *why is a data piece in the result*? Nevertheless, the approach does not draw a history of the database regardless of format. It takes the database description as a starting point to move from one backward "shot" to another. According to Archer (2011), "current [data] provenance models store with each data item only the single provenance expression that ties that item to its immediate parents… [and] provide no language to easily re-assemble these generations so that the entire history of a data item may be queried." Alternatively, FM provides an avenue to a high-level conceptual depiction of the complete history of events that led to producing a particular piece of data, as shown in the FM representation of Figure 8.

In Figure 8, two pieces of data, Compound and Formula, are input (circles 1 and 2) to create (3) a tuple (4) that flows to the Table sphere (5). (Note that, for simplicity, the *Tuple sphere* is not enclosed in a box.) Accordingly, in the Table sphere, processing of the resultant tuple (6) and the previous version of the table (7) *creates* a new Table (8). Thus, we have two pieces of data (1 and 2) constructing a new tuple (3); this new tuple and the stored table (7) produce a new table (8) with the new tuple inserted. The new table is again stored (9). Events can be tracked as follows:

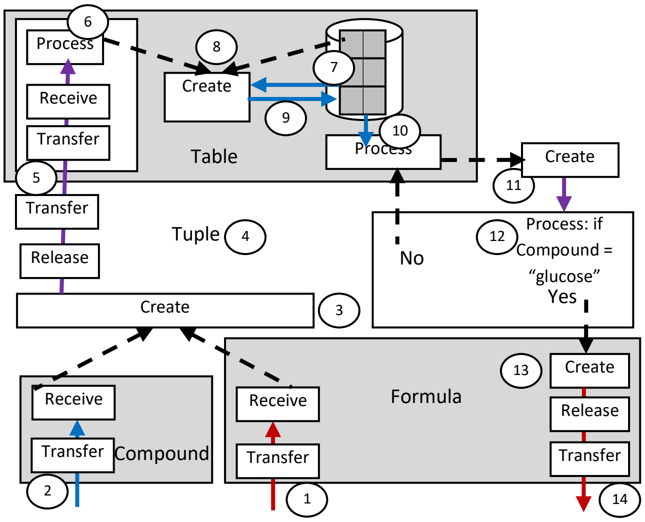


Figure 8. Complete history of a value for Formula.

Glucose and C6H12O6 → (*Glucose* , *C6H12O6*) → OLD table + (*Glucose* , *C6H12O6*) → NEW Table

This sequence represents a complete conceptual (implementation-independent) description of the recorded pre-query history. The granularity of data provenance (e.g., all or some columns or attributes), i.e., the degree of fineness to which data-field histories are monitored, is a policy decision.

At time of query, the table is retrieved (7), not for the purpose of updating it with a new tuple, but to process the query. Thus, the table is processed (10) to produce tuples (11) one after another to check if Compound = "glucose" (12), and if so, the Formula value is extracted (13) from the tuple and output (14).

Figure 9 shows possible metadata that can be collected in this example.

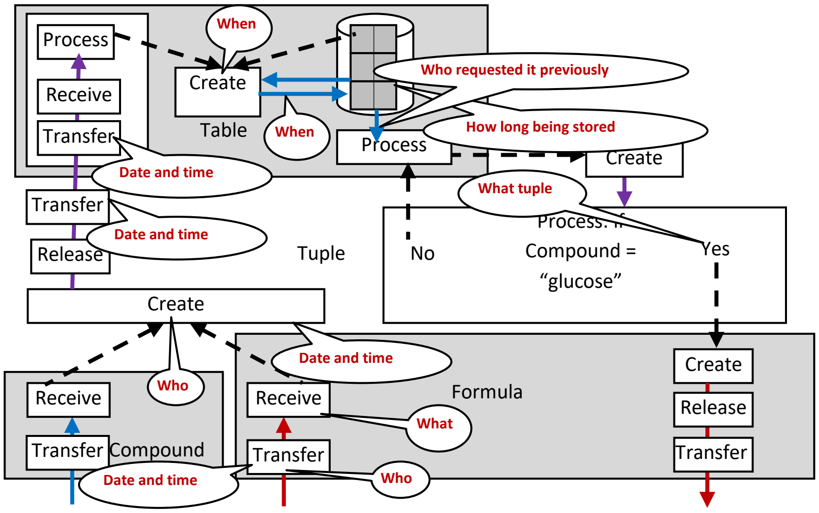


Figure 9. Possible metadata.

The contrast with Müller's Figure 7 reminds us of a scientist producing table after table of observations while leaving it to the reader to configure a chronology of events of input, intermediate constructs, resultant constructs, divisions of these, etc. Alternatively, FM presents a reasonably exact representation of these events that replaces or complements the table-based chronology of events. Figure 10 shows possible "actors" who participate in generating the data items in the example. These actors may contribute to the metadata, such as data identifying their roles.

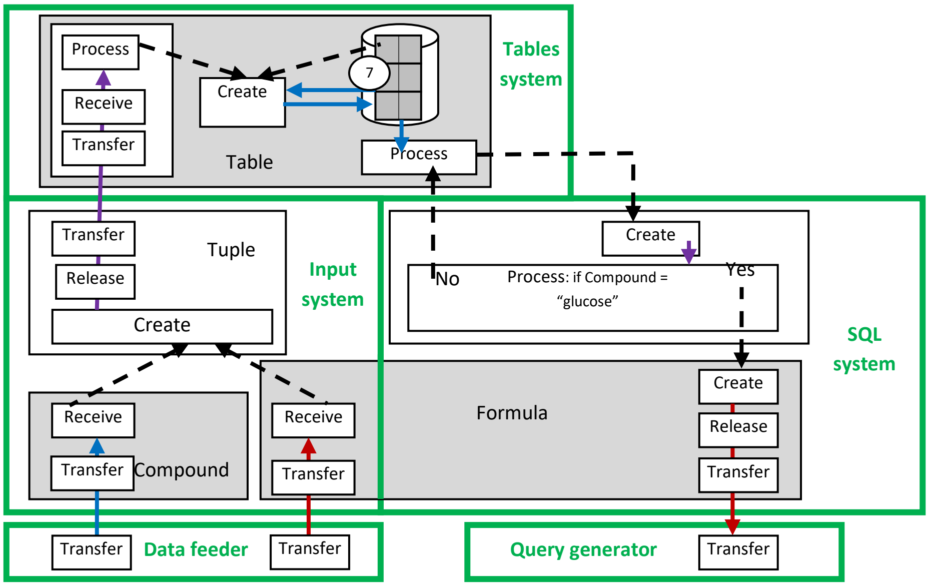


Figure 10. Different actors participate in generating the data items in the example.

Provenance of a Cake

According to Cheney (2010), OPM lacks "semantics" in the operational sense, which motivates investigation of the use of structural causal models (see the next example) as semantics for these graphs. Cheney (2010) gives the OPM graph shown in Figure 11 of the "provenance of a cake." Figure 12 shows the corresponding structural causal model. Without loss of generality, and for simplicity, we consider only sugar and flour as ingredients in the cake.

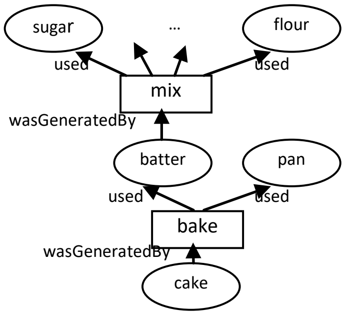


Figure 11. OPM graph. Ovals denote "artifacts" while boxes denote "processes" (redrawn, partial from (Cheney, 2010)).

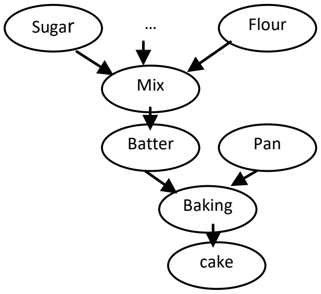


Figure 12. Structural causal model, also depicted as a graph (redrawn, partial from (Cheney, 2010)).

FM Representation

Figure 13 shows the FM representation of the "provenance of a cake." Sugar, flour, mix, batter, and cake are things; mixing, beating, and baking are types of processing; and pan and batter are spheres. The cake (circle 1) is created by baking (2) batter that has flowed (conceptually) along with a pan (3) after being beaten (4) while in the pan (note the intersection of the pan and batter spoon). Before beating, a mix was created (5) by combining (6) sugar and flour in the pan. For simplicity, the authors opt not to distinguish between the pan and the cake in the oven but treat them as one unit (remember the example of car and number of passengers in section 2). Nevertheless, it is possible to represent each separately if desired. Furthermore, note how the *physical pan* comprises conceptual sub-spheres of pan, batter, and mix. The mix flows to the conceptual sub-sphere intersecting with the *batter spoon* sphere (white box in Figure 13 labeled Mix, including the purple intersection).

Note also how OPM processes are modeled in FM. Figure 14 presents a comparison of some parts of the diagrams in OPM and FM. This shows the OPM to be a kind of "shorthand**"** diagramming method or a technique of rapid inscription by means of abbreviations, symbols, and notations. In further comparison with the OPM notions, we see the following.

- Mix *wasGeneratedBy* Batter (Figure 11) corresponds to *Create* in FM (circle 5 in Figure 14).
- *Used* (Figure 11) corresponds to (1) flow of materials to create a mix, and (2) the intersection of the pan and the process of beating the batter (see lower right of Figure 14). Note that such an intersection is conceptual: the mix is simultaneously in the pan *sphere* and the batter *sphere*.

The notion of flow in FM interweaves all events: the sugar and flour that become a mix that is beaten in the pan to process into batter that is placed in the oven to create a cake. As can be seen, the FM representation, based on the notion of flow, forces the modeler to: ”connect the dots” and paint a complete chronology of events in the system.

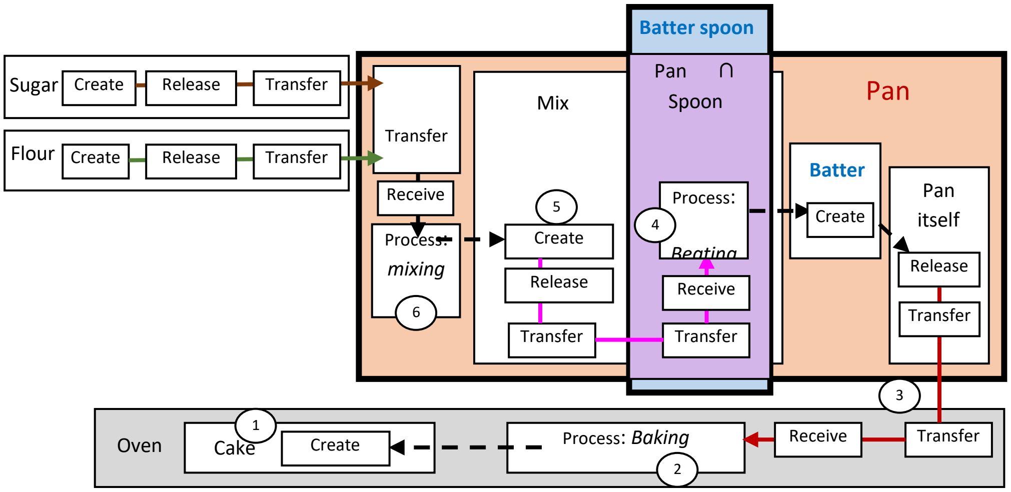
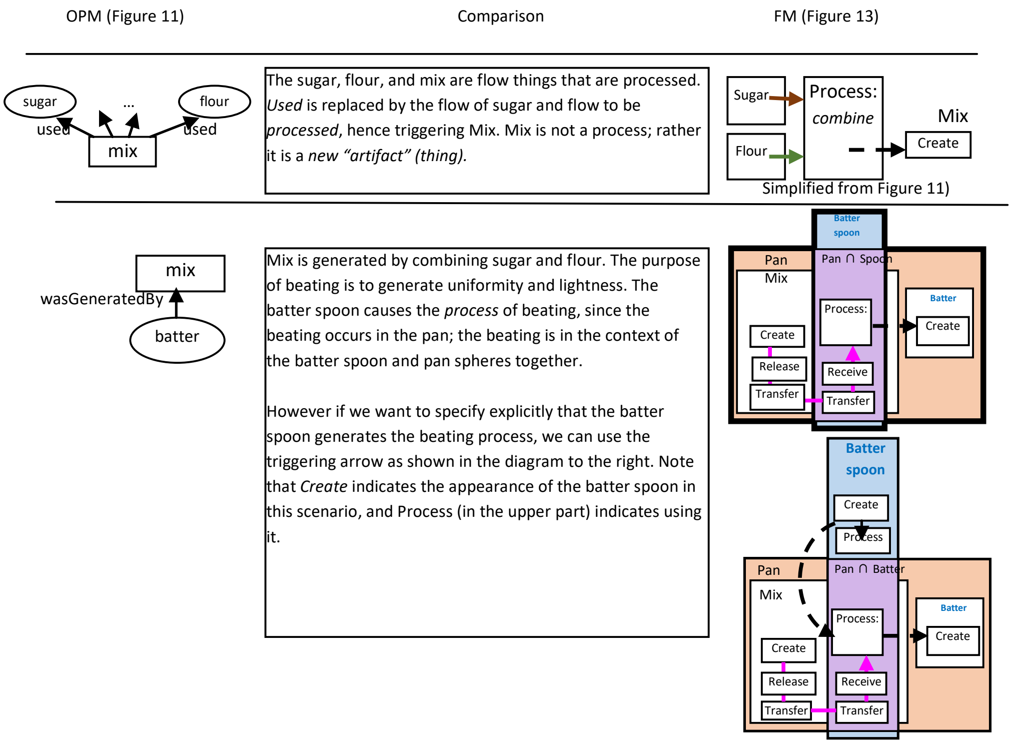


Figure 13. FM representation of the "provenance of a cake."

Provenance and Queries

Many agents can routinely implement baking a cake as represented in FM; nevertheless, cakes produced by different agents are likely to be different from each other within each baking attempt. Accordingly, deviation in quality of the cake must be understood through details of provenance acquired by querying different implementations. Much metadata can be collected about each production phase of a cake. For simplicity we will concentrate on the particular metadata of producing cakes:



The sugar, flour, and mix are flow things that are processed. *Used* is replaced by the flow of sugar and flow to be *processed*, hence triggering Mix. Mix is not a process; rather it is a *new "artifact" (thing).*

Mix is generated by combining sugar and flour. The purpose of beating is to generate uniformity and lightness. The batter spoon causes the *process* of beating, since the beating occurs in the pan; the beating is in the context of the batter spoon and pan spheres together.

However if we want to specify explicitly that the batter spoon generates the beating process, we can use the triggering arrow as shown in the diagram to the right. Note that *Create* indicates the appearance of the batter spoon in this scenario, and Process (in the upper part) indicates using it.

Figure 14. Comparison of some parts of the OPM graph with FM.

(1) The amount of sugar used
(2) The amount of flour used
(3) The time taken to beat the mixture

Accordingly, the authors enrich the FM representation with related spheres as shown in Figure 15, indicated by circles 1, 2, and 3. In the sphere of circle 3, the start and end times of beating are recorded.

Modeling a particular instance of producing a cake needs additional specification, suitable for dynamic aspects of the static FM representation, such as shown in Figure 16. Specifically, this phase focuses on the event-oriented instance of the method of producing the cake. We need to embed operational semantics that specify the chronology of events for a particular implementation.

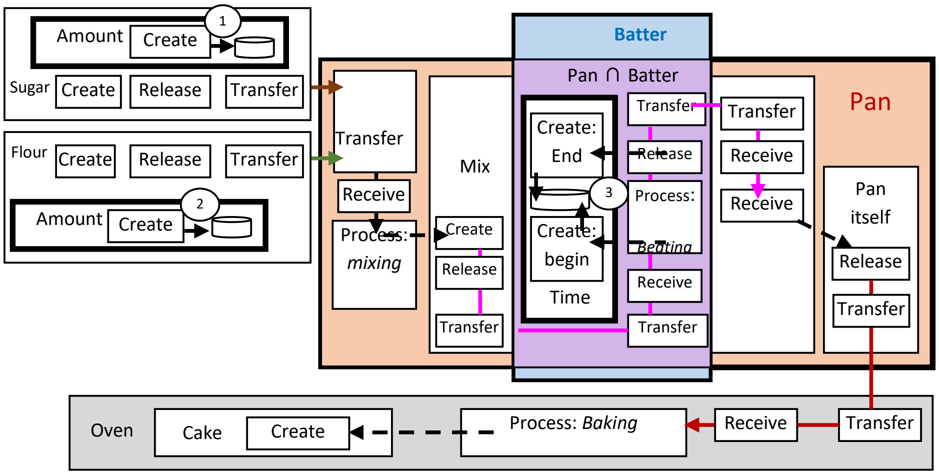


Figure 15. FM representation of the "provenance of a cake" where amounts of sugar and flour, and beating time are recorded as metadata (dark boxes).

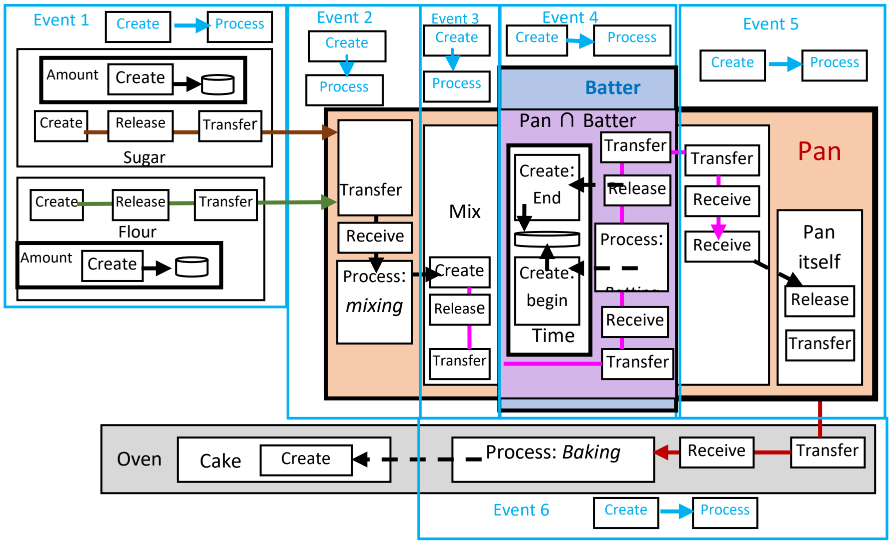


Figure 16. Events of execution of a particular cake, with amounts of sugar and flour, and beating time recorded as metadata.

Events (happenings, occurrences) are *things* that can be created, processed, released, transferred, and received. Accordingly, an operational semantics can be defined to describe scheduling of events in the FM representation, thereby specifying a thread of control of execution. Note that the flow and triggering force a partial order on some events, e.g., *release* is preceded either by *create*, *process*, or *receive*. This sequential ordering is similar to the actual execution of a computer program in which a specific execution uncovers concurrency between threads of events offering several options for executions or occurrences.

Even though each stage in the spheres of FM representation can be activated by a separate event, for simplicity's sake we declare 6 non-atomic events in the sequence of baking a cake, as shown in Figure 16. Each event is created and processed, i.e., takes its course:

Event 1 activates preparing the sugar and flour
Event 2 activates mixing

Event 3 activates creating the final mix
Event 4 activates beating into batter
Event 5 activates the final batter in the pan and moving the pan into the oven
Event 6 activates baking

Of course, these events can be included in the metadata, e.g., a user may have to wait sometime after event 1 to activate event 2.

Accordingly, in our example, production of a specific cake (cake No. 1234) was executed (past tense) according to the operational semantics of Figure 16, and its metadata were recorded— the amounts of sugar and flour used, and the amount of time the batter was beaten—for use in understanding the subsequent quality of that particular cake.

As can be seen, the same FM representation was used to model the description of steps involved in making a cake, as well as to specify the required metadata and operational semantics of events.

PROVENANCE OF A COFFEE SHOP

Kwasnikowska, Moreau, and Van den Bussche (2015) offer an OPM diagram of the following scenario:

> Alice and her young son Bob ordered a latte and a fruit juice in a coffee shop… Alice, who could observe the activities behind the counter, identified three different *processes*. The cashier took the order and associated payment. As soon as the order was taken, the cashier put an empty cup on a tray next to the coffee machine; once payment was taken, the cashier added a till receipt to the same tray. The coffee machine operator picked up the cup, and filled it with the requested coffee, as per receipt, and handed the tray over to Alice. A third person behind the counter served other drinks, on request from the cashier. Alice was unable to ascertain how information was communicated (e.g., the request was stated by cashier, or order read from receipt); what is definite from Alice's viewpoint is the juice was also delivered with the tray.

Accordingly, its OPM graph (Figure 17) is developed to consist of nodes as artifacts and processes, along with edges. Artifacts consist of an "order", "cash", an "empty cup", a "receipt", a "juice" and a "latte". For Bob, there is only the process "Get Drink". Alice's events involve three processes: "Take Order", "Make Coffee", and "Provide other Beverages". Bob's version of events is represented in the same graph as Alice's version.

In Figure 17, these descriptions are distinguished by color (black for Alice's version, and violet for Bob's; artifacts "order" and "juice" belong to both versions.). In OPM, an edge source represents an effect and an edge represents a cause. Four types of edges can be distinguished depending on type of cause and effect: a used-edge is between a process and an artifact; a generated-by edge is between an artifact and a process; a derived-from edge is between two artifacts; and an informed-by edge is between two processes (Kwasnikowska et al., 2015).

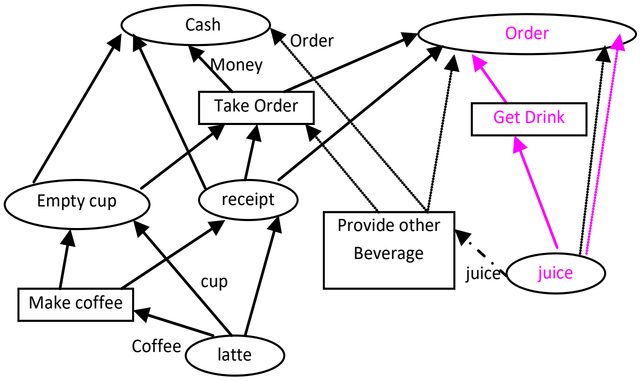


Figure 17. OPM graph (redrawn, partial from (Kwasnikowska et al., 2015)).

The aim of this summarized account and the partial drawing of the OPM graph is not to give a fair description of this approach, but rather to allow the reader to develop a general opinion about the methodology and its effectiveness as a model of a real situation. For a complete description and discussion of the given OPM graph and the sequence of events, the reader is referred to the original source (Kwasnikowska et al., 2015). At this stage of development of using diagrams in the field of provenance, we will utilize this attempt at *human understanding* of the OPM graph as a base for contrasting it with the FM representation; hence, it is hoped to provide justification for its proposed use as an alternative to OPM graphs.

Figure 18 shows the FM representation of the coffee shop scenario.

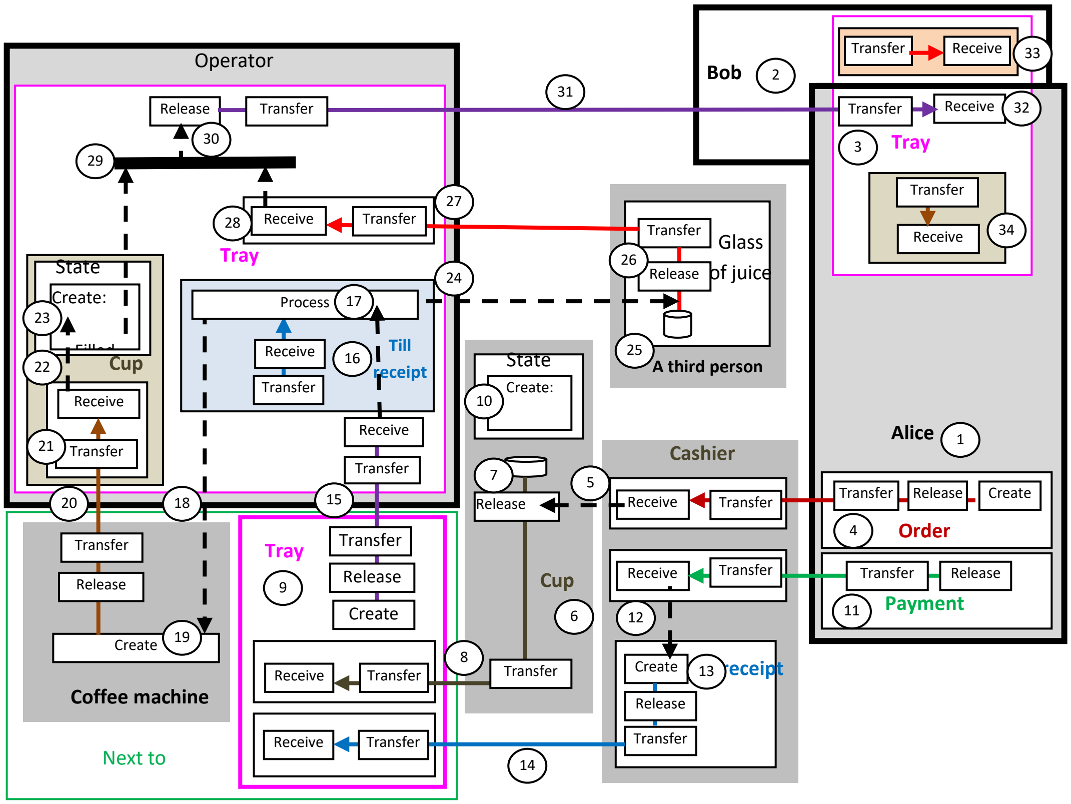


Figure 18. FM representation of coffee shop order by Alice and Bob.

On the right, we see the spheres of Alice (circle 1) and Bob (2) that include the tray (3) as a shared sub-sphere between them, where the coffee belongs to Alice and the juice belongs to Bob. This detail (of *one* tray) is forced by the nature of the *flow* in FM representation, which requires *continuity* in the movement of things, analogous to modeling liquid flow in the pipes that carry fluids. However, it is possible to draw the situation such that the coffee and juice are on two trays, or in containers without a tray. The textual description given by Kwasnikowska et. al (2015) does not specify this detail, but the FM flows require filling such descriptive gap in the modeled situation. This problem does not arise in the OPM graph because that model reflects discrete "shorthand" notations of events, like a movie where we see the hero in Paris, on the Eiffel Tower, then in the next shot in London, climbing Big Ben, leaving it to viewers to fill in the gap during which he traveled between the two cities.

In FM, such travel must be accounted for, represented by a flow (arrow) from Paris to London or by some triggering mechanism. Continuing to describe the figure,

- Alice gives an order (4) that flows to the cashier. This triggers (5) the cashier to immediately take a (physical) cup (6) (for simplicity, the cup flow system is not outlined by a box) from the stock of cups (7) and set it (8) on the tray (9). Note that the cup is empty (its state - 10).
- Alice also makes payment to the cashier (11). The payment triggers (12) the cashier to create (13) a till receipt that he puts on the tray (cup flows to, 14). We assume that the tray is available next to the stock of cups. If this is not the case then it is possible to depict taking the tray from its stock. The textual description given by Kwasnikowska et al. (2015) does not specify this detail. This point is raised because the nature of flow in the FM representation demands specifying the source of the flow of the tray.
- The tray flows to the coffee machine operator (15). Note that the flow here does not necessarily mean physical movement. It indicates that the tray has entered the sphere of the operator (intention – responsibility – his agency). As discussed previously in the example of a car and its passengers, this means implicitly that everything on the tray is also in the sphere of the coffee machine operator. Accordingly, the arrival of the tray triggers (16) processing of the till receipt (17), triggering (18) creation (19) of a flow of coffee from the coffee machine into the cup on the tray (20). Again, the previous flow of the tray (15) means that the cup has also moved with it (21). This flow of coffee from the machine continues until it triggers (22) a visual sign that the cup is full (23) and the operator responds accordingly.
- Additionally, as a result of processing the till receipt (17), the operator triggers (24) a third person (25) to release a glass of juice (26) that flows to the tray (27). How the operator communicates with the third person (physically gives the till receipt, verbally, some kind of signal) is not known, hence, a triggering mechanism is used (Müller, 2016). In addition, it is not known if the glass is another cup or taken from a stock of glasses, as was the case with the cup. These details can be modeled when they are known.
- Note that these repeatedly noted missing details such as *one or two trays* and *cup or glass of juice*, are easily emended to the description if Alice is recording her observation in the FM diagrammatic language instead of English text. The FM language converts an observer's description of events to an engineering-like drawing.
- Now the glass of juice has arrived on the tray (28), while the cup of coffee is full (23); these two conditions concurrently (29) trigger (30) the flow of the tray to Alice and Bob (31). The thick line (29) indicates the realization of both occurrences triggering. It is a familiar computer science synchronization used for simplicity, but it can be replaced by FM diagram notions.
- Finally, the tray is received by Alice and Bob (32), with juice belonging to Bob (33) and coffee belonging to Alice (34).

Figure 18 is a particular instance of coffee shop service. To see its general model with recording of metadata, we can replace Alice and Bob with a general customer. Additionally, \ the work quality of the coffee machine operator can be examined by recording the number of times the delivered tray fails to include the exact items ordered by the customer.

Accordingly, Fig 19 is a modification of Figure 18 to accomplish such monitoring, as follows:

- A list of item descriptions has been added, explicitly, as a separate sub-sphere of the till receipt (circle 1), in case the till receipts include only numbers or codes of ordered items
- A manager sphere is added (2) as the sphere of a person who receives the tray before delivery to the customer (3) to compare the content of the tray with the items listed on the till receipt (4) to create information (5) that is processed (6) by the manager to trigger (7) either:
  1. passing the tray to the customer (8 - correct content), or
  2. finding something wrong, triggering (9) recording of this instance of provenance (operator, type of error, time, …)

As can be seen, the FM representation lends itself to modeling the system and provenance, let alone the operational semantics, in a uniform way.

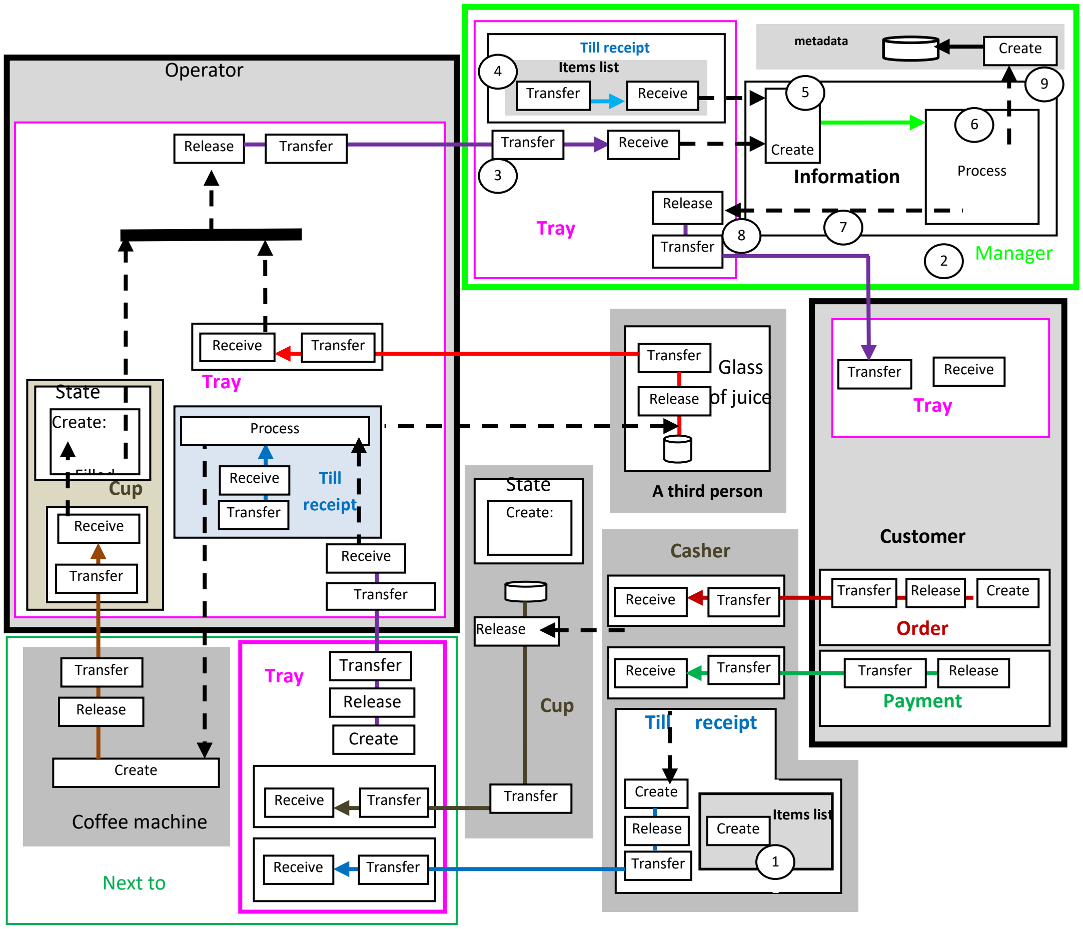


Figure 19. FM representation with a sample provenance.

Conclusion

Research in this paper examines proposed provenance representation models, including standard models that can be used for capturing and publishing provenance such as the Open Provenance Model (OPM). Without loss of generality, the focus of this paper is on OPM, specifically its graph representations. The paper proposes an alternative flow-based diagrammatic language that can form the foundation for modeling provenance.

At this stage of development of diagrams for use in the field of provenance, contrasting the two methodologies is based on *human evaluation* of their expressiveness. Accordingly, several cases of OPM graphs have been re-depicted in terms of the proposed language. The results seem to demonstrate that the proposed model is a viable alternative in the area of diagrammatic representation for provenance applications. Nevertheless, OPM graphs have been studied fairly extensively, resulting in development of a formal foundation, a background that FM lacks. Future work can be conducted in this area by formalizing the FM model in order to define it more precisely and allow more accurate comparisons with other models.

References

Acar, U.A., Ahmed, A., Cheney, J., & Perera, R. (2013). A core calculus for provenance. *Journal of Computer Security*, *21*(6), 919-969. doi:10.3233/JCS-130487

Agrawal, P., Benjelloun, O., Das Sarma, A., Hayworth, C., Nabar, S., Sugihara, T., & Widom, J. (2006, September). Trio: A system for data, uncertainty, and lineage. *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea,* 1151-1154.

Al-Fedaghi, S. (2016a, August). Thing-oriented learning: application to mathematical objects. *Proceedings of the 19th IEEE International Conference on Computational Science and Engineering (CSE 2016)*, *Paris, France.*

Al-Fedaghi, S. (2016b, October). Activity recognition and sensor positioning. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2016)*, *Budapest, Hungary.*

Al-Fedaghi, S. (2016c). Conceptual modeling in simulation: A representation that assimilates events. *International Journal of Advanced Computer Science and Applications, 7*(10), 281-289.

Al-Fedaghi, S. (2016d). Toward a philosophy of data for database systems design. *International Journal of Database Theory and Application, 9*(10), 47-62.

Al-Fedaghi, S. (2016e). Diagrammatic modeling language for conceptual design of technical systems: A way to achieve creativity. *International Review of Automatic Control, 9*(4), 252-258.

Al-Fedaghi, S. (2017a, February). Thinking in terms of flow in design of software systems. *Proceedings of the Second International Conference on Design Engineering and Science (ICDES 2017)*, *Kortrijk , Belgium.*

Al-Fedaghi, S. (2017b, March). Sets with members as machines with things that flow. *Proceedings of the the International Conference on Internet of Things, Data and Cloud Computing (ICC 2017)*, *University of Cambridge, UK.*

Archer, D. W. (2011). *Conceptual modeling of data with provenance.* Unpublished doctoral dissertation, Portland State University, OR.

Buneman, P., Khanna, S., & Tan, W.-C. (2000). Data provenance: Some basic issues. *Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science, London, UK*, 87-93.

Cheney, J. (2010). Causality and the semantics of provenance. In S. B. Cooper, E. Kashefi, and P. Panangaden (Eds.), *Proceedings Sixth Workshop on Developments in Computational Models: Causality, Computation, and Physics (DCM 2010),* 63-74. doi:10.4204/EPTCS.26.6

Cheney, J., Chiticariu, L., & Tan, W.-C. (2007). Provenance in databases: Why, how, and where. *Foundations and Trends in Databases, 1*(4).

Cheney, J., Chong, S., Foster, N., Seltzer, M., & Vansummeren, S. (2009, October). Provenance: A future history. *ACM OOPSLA Conference*, *Orlando, FL,* 957-964.

Cohen, E. (1999). From ugly duckling to swan: Reconceptualizing information systems as a field of the discipline informing science. *Journal of Computing and Information Technology, 7*(3), 213-219.

Cohen, E. (2009). A philosophy of informing science. *Informing Science: The International Journal of an Emerging Transdiscipline, 12*, 1–15. Retrieved October 16, 2009 from http://inform.nu/Articles/Vol12/ISJv12p001-015Cohen399.pdf

Cuevas-Vicenttín, V., Ludäscher, B., Missier, P., Belhajjame, K., Chirigati, F., Wei, Y., … Cao, Y. (2016, May). ProvONE: A PROV extension data model for scientific workflow provenance, Version 1 Draft, DataONE Cyberinfrastructure Working Group. http://jenkins-1.dataone.org/jenkins/view/Documentation%20Projects/job/ProvONE-Documentation-trunk/ws/provenance/ProvONE/v1/provone.html#bib-MCF+11

Cui, Y., Widom, J., & Wiener, J. L. (2000). Tracing the lineage of view data in a warehousing environment. *ACM Transactions on Database Systems, 25*(2), 179-227.

Gill, T. G., & Bhattacherjee, A. (2009). Informing science at a crossroads: The role of the client. In T. G. Gill & E. Cohen (Eds.), *Foundations of Informing Science: 1999-2008* (pp. 21-55). Santa Rosa, CA: Informing Science Press.

Green, T. J., Karvounarakis, G., & Tannen, V. (2007, June). Provenance semirings, *PODS '07*, *Beijing, China,* 31-40. Retrieved from http://users.ics.forth.gr/~gregkar/publications/pods2007.pdf

Green, T. J., Karvounarakis, G., Taylor, N. E., Biton, O., Ives, Z. G., & Tannen, V. (2007). Orchestra: Facilitating collaborative data sharing. *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pp. 1131-1133.

Ikeda, R., & Widom, J. (2010). Panda: A system for provenance and data. *Proceedings of the 2nd Conference on Theory and Practice of Provenance, TAPP'10*, *Berkeley, CA.*

Kwasnikowska, N., Moreau, L., & Van den Bussche, J. (2015, May). A formal account of the Open Provenance Model. *ACM Transactions on the Web, 9*(2).

Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P. T., … Van den Bussche, J. (2011). The Open Provenance Model core specification (v1.1). *Future Generation Computing Systems, 27*(6), 743-756.

Müller, T. (2016, September). Have your cake and eat it, too: Data provenance for TuringComplete SQL queries. *Proceedings of the VLDB 2016 PhD Workshop*, *New Delhi, India.*

Nassopoulos, S., Serrano-Alvarado, P., Molli, P., & Desmontils, E. (2012). D.3.2: State of the art of purpose-based, usage control approaches [Technical Report], LINA-University of Nantes. 2015. hal-01174210

Nurse, J. R. C., Rahman, S. S., Creese, S., Goldsmith, M., & Lamberts, K. (2011, May). Information quality and trustworthiness: A topical state-of-the-art review. *2011 International Conference on Computer Applications and Network Security (ICCANS 2011)*, *Maldives.*

Ram, S., & Liu, J. (2009). A new perspective on semantics of data provenance. *Proceedings of the First International Conference on Semantic Web in Provenance Management, 526*, 35-40.

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Urbana, IL: University of Illinois Press.

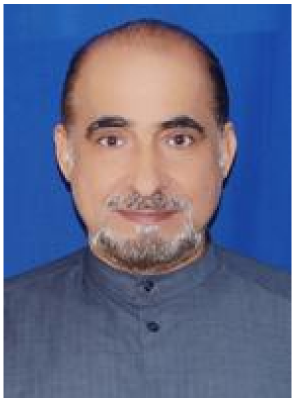
Stanford Encyclopedia of Philosophy, Philosophy of History. (2012). http://plato.stanford.edu/entries/history/

Travica, B. (2014). Think process, think in time: Advancing study of informing systems. *Informing Science: the International Journal of an Emerging Transdiscipline, 17*, 133-148. Retrieved from http://www.inform.nu/Articles/Vol17/ISJv17p133-148Travica0519.pdf

W3C (World Wide Web Consortium). (2013). *PROV: Overview. An Overview of the PROV Family of Documents.* Retrieved from http://www.w3.org/TR/prov-overview/

W3C (World Wide Web Consortium). (2016). *Provenance current status.* Retrieved from https://www.w3.org/standards/techs/provenance#w3c_all

Biography



Dr. Sabah Al-Fedaghi holds an MS and a PhD in computer science from the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois, and a BS in Engineering Science from Arizona State University, Tempe. He has published two books and more than 260 papers in journals and conferences on software engineering, database systems, information systems, computer/ information privacy, security and assurance, information warfare, and conceptual modeling. He is an associate professor in the Computer Engineering Department, Kuwait University. He previously worked as a programmer at the Kuwait Oil Company and headed the Electrical and Computer Engineering Department (1991–1994) and the Computer Engineering Department (2000–2007).