Troy Costa Kohwalter

<TÍTULO DO TRABALHO>

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Computação Visual.

Advisors: Prof. Dr. Esteban G. W. Clua

Prof. Dr. Leonardo G. P. Murta

Niterói

2013

Ficha Catalográfica – Esta página deve ser removida na versão a ser entregue para a banca, mas deve ser reinserida na versão final, com a ficha catalográfica fornecida pela biblioteca. Informações sobre este processo devem ser obtidas na secretaria da pós-graduação.

Clique aqui para digitar texto.

Clique aqui para digitar texto.

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Computação Visual.

Aprovada em <MES> de <ANO>.

BANCA EXAMINADORA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. <NOME DO ORIENTADOR> – Orientador

UFF

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. <NOME DO AVALIADOR>

<INSTITUIÇÃO DO AVALIADOR>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. <NOME DO AVALIADOR>

<INSTITUIÇÃO DO AVALIADOR>

Niterói

<ANO>

"Dedicatória(s): Elemento opcional onde o autor presta homenagem ou dedica seu trabalho" (ABNT, 2005).

**Agradecimentos**

"Elemento opcional, colocado após a dedicatória" (ABNT, 2005).

"Epígrafe: Folha onde o autor apresenta uma citação, seguida de indicação de autoria, relacionada com a matéria tratada no corpo do trabalho." (ABNT, 2005).

**Resumo**

"Elemento obrigatório, constituído de uma sequência de frases concisas e objetivas e não de uma simples enumeração de tópicos, não ultrapassando 500 palavras" (ABNT, 2005).

Palavras-chave: "Palavras representativas do conteúdo do trabalho, isto é, palavras-chave e/ou descritores, conforme a ABNT NBR 6028" (ABNT, 2005).

**Abstract**

"Elemento obrigatório, em língua estrangeira, com as mesmas características do resumo em língua vernácula" (ABNT, 2005).

Keywords: "Palavras representativas do conteúdo do trabalho, isto é, palavras-chave e/ou descritores, na língua" (ABNT, 2005).

**Lista de Ilustrações**

[Figure 1: Edges in OPM. Source: (MOREAU *et al.*, 2007). 20](#_Toc353465411)

[Figure 2: Artifact introduction and elimination. Source: (MOREAU *et al.*, 2007). 23](#_Toc353465412)

[Figure 3: Process introduction. Source: (MOREAU *et al.*, 2007). 23](#_Toc353465413)

[Figure 4: Inference. Source: (MOREAU *et al.*, 2007). 24](#_Toc353465414)

[Figure 5: OPM’s Layered Architecture. Source: (MOREAU *et al.*, 2007) 24](#_Toc353465415)

[Figure 6: PROV organization. Source: (GROTH; MOREAU, 2010) 25](#_Toc353465416)

[Figure 7: PROV Entities and relations. Source: (GIL; MILES, 2010) 27](#_Toc353465417)

[Figure 8: Using *Expanded Relations*. Source: (GIL; MILES, 2010) 29](#_Toc353465418)

[Figure 9: *Optional ID* and *Further Relations*. Source: (GIL; MILES, 2010) 29](#_Toc353465419)

[Figure 10: Time information. Source: (GIL; MILES, 2010) 31](#_Toc353465420)

[Figure 11: Data model diagram. Gray classes represent provenance classes. 38](#_Toc353465421)

[Figure 12: Example of structure 40](#_Toc353465422)

[Figure 13: Relationship between a game using the framework and *Proof Viewer*. 42](#_Toc353465423)

[Figure 14: Example of a generated provenance graph. 43](#_Toc353465424)

[Figure 15: Collapsing Nodes 44](#_Toc353465425)

**Lista de tabelas**

[Table 1: PROV optional attributes 30](#_Toc353266335)

[Table 2: OPM x PROV 34](#_Toc353266336)

**Lista de abreviaturas e siglas**

"Elemento opcional, que consiste na relação alfabética das abreviaturas e siglas utilizadas no texto, seguidas das palavras ou expressões correspondentes grafadas por extenso. Recomenda-se a elaboração de lista própria para cada tipo" (ABNT, 2005).

HP – Hit points

NPC – Non player character

OPM – Open Provenance Model

IPAW – International Provenance and Annotation Workshop

RPG – Role-playing game

HCI – Human-Computer Interaction

TRUE – Tracking Real-Time User Experience

UIE – User Initiated Events

CMDS - Classical Multidimensional Scaling

**Lista de símbolos**

"Elemento opcional, que deve ser elaborado de acordo com a ordem apresentada no texto, com o devido significado" (ABNT, 2005).

Contents

[Chapter 1 – Introduction 16](#_Toc354161705)

[1.1 Motivation 16](#_Toc354161706)

[1.2 Goals 16](#_Toc354161707)

[1.3 Research Questions 16](#_Toc354161708)

[1.4 Contributions 16](#_Toc354161709)

[1.5 Organization 16](#_Toc354161710)

[Chapter 2 – Game Flow Analysis 17](#_Toc354161711)

[2.1 Introduction 17](#_Toc354161712)

[2.2 Usage 17](#_Toc354161713)

[2.3 UNDECIDED TITLE 17](#_Toc354161714)

[2.4 Related Work 17](#_Toc354161715)

[2.5 Final Considerations 17](#_Toc354161716)

[Chapter 3 – Provenance 18](#_Toc354161717)

[3.1 Introduction 18](#_Toc354161718)

[3.2 Open Provenance Model 19](#_Toc354161719)

[3.2.1 Types and Relations 20](#_Toc354161720)

[3.2.2 Time Information 21](#_Toc354161721)

[3.2.3 Completion Rules 22](#_Toc354161722)

[3.3 PROV 25](#_Toc354161723)

[3.3.1 Types and Notations 26](#_Toc354161724)

[3.3.2 Further Notations 28](#_Toc354161725)

[3.3.3 Time Information 30](#_Toc354161726)

[3.3.4 Inference 31](#_Toc354161727)

[3.4 Comparison Between Models 33](#_Toc354161728)

[3.5 Final Considerations 34](#_Toc354161729)

[Chapter 4 – Provenance in Games 35](#_Toc354161730)

[4.1 Introduction 35](#_Toc354161731)

[4.2 Data Model 36](#_Toc354161732)

[4.3 Provenance Gathering and Data Structure 39](#_Toc354161733)

[4.4 Provenance Visualization 41](#_Toc354161734)

[4.4.1 Shape and Color 43](#_Toc354161735)

[4.4.2 Filters 44](#_Toc354161736)

[4.4.3 Layouts 46](#_Toc354161737)

[4.5 Final Considerations 46](#_Toc354161738)

[Chapter 5 – Implementation 47](#_Toc354161739)

[5.1 Introduction 47](#_Toc354161740)

[5.2 SDM 47](#_Toc354161741)

[5.3 Guiding Example 47](#_Toc354161742)

[5.4 Provenance Gathering 47](#_Toc354161743)

[5.4.1 Information Storage 47](#_Toc354161744)

[5.5 Provenance Analysis 47](#_Toc354161745)

[5.5.1 Node Representations 47](#_Toc354161746)

[5.5.2 Edge Representations 47](#_Toc354161747)

[5.5.3 Node Filters 47](#_Toc354161748)

[5.5.4 Edge Filters 47](#_Toc354161749)

[5.6 Final Considerations 47](#_Toc354161750)

[Chapter 6 – Evaluation 48](#_Toc354161751)

[6.1 Introduction 48](#_Toc354161752)

[6.2 Experiment Planning 48](#_Toc354161753)

[6.3 Experiment Execution 48](#_Toc354161754)

[6.4 Statistical Analysis 48](#_Toc354161755)

[6.5 Threats to Validity 48](#_Toc354161756)

[6.6 Final Considerations 48](#_Toc354161757)

[Chapter 7 – Conclusion 49](#_Toc354161758)

[7.1 Contributions 49](#_Toc354161759)

[7.2 Limitations 49](#_Toc354161760)

[7.3 Future Work 49](#_Toc354161761)

[References 50](#_Toc354161762)

[Glossary 53](#_Toc354161763)

[Appendix A – Título do Apêndice 54](#_Toc354161764)

[Annex A – Título do Anexo 55](#_Toc354161765)

[Index 56](#_Toc354161766)

# – Introduction

## Motivation

## **Goals**

## **Research Questions**

## Contributions

## Organization

# – Game Flow Analysis

## Introduction

## Usage

## **UNDECIDED TITLE**

## Related Work

## Final Considerations

# – Provenance

## Introduction

Results of scientific experiments cannot be understood without the knowledge of the meaning of data and circumstances occurred during their creation. This type of knowledge includes data provenance (DAVIDSON; FREIRE, 2008; FREIRE *et al.*, 2008). Provenance is well understood in the context of art or digital libraries where historical documentation refers to an object’s life cycle (PREMIS WORKING GROUP, 2005). However, for digital provenance there are two types of provenance perspectives: retrospective and prospective (FREIRE *et al.*, 2008). The prospective provenance focuses on specifications and the necessary steps to achieve the generated data, while retrospective provenance focuses on the executed steps and external information used to derive the data.

Recently, data provenance in scientific experimentation has become an important topic in scientific research and, as consequence, workshops and conferences for the subject were specifically created (SIMMHAN; PLALE; GANNON, 2005). The *International Provenance and Annotation Workshop* (IPAW) (MOREAU *et al.*, 2002) was one of the first data provenance workshops to be created. In each edition, the scientific community lists challenges of data provenance to be solved and receives scientific works with possible solutions. During IPAW’06, participants were interested in questions about provenance for the usage in digital data, involving topics related to documentation, data annotation, and data derivations (BOSE; FOSTER; MOREAU, 2006). As a result, the first model of digital provenance, the *Open Provenance Model* (OPM) (MOREAU *et al.*, 2007), was created. The OPM has been designed to address the issues raised during the *Provenance Challenge* (MILES *et al.*, 2010), such as understanding provenance systems and how to express provenance information. It is based on retrospective provenance.

Later, another provenance model, PROV (GIL; MILES, 2010), was developed by the provenance incubator group at W3C (GIL *et al.*, 2009). According to the group, provenance of digital objects represents the object’s origins and PROV is a proposed specification to represent these provenance records. These records contain descriptions of the entities and activities involved in producing and delivering or otherwise influencing a given object. PROV is also focused on retrospective provenance, similarly to OPM. The usage of provenance, regardless of the model, provides a critical foundation for assessing the authenticity of data, enabling reliability and reproducibility, and is a crucial component of workflow systems (GIL *et al.*, 2007; GROTH; MOREAU, 2010).

When PROV was proposed, the OPM model was already being used in several approaches. However, the fact that PROV is supported by the W3C makes the possibility of becoming the default provenance model and the migration from OPM to PROV a possibility in the near future. Nevertheless, the aim of this chapter is to present a study of both digital provenance models, which uses retrospective provenance, as well as comparing these models, pointing out their similarities and differences.

As such, this chapter is organized as follow: Section 3.2 provides a guiding example to aid understanding of provenance relationships in a provenance graph. Section 3.3 and 3.4 describe the OPM and PROV, respectively. Section 3.5 compares both digital provenance models and, lastly, section 3.6 presents the final considerations of this chapter.

## Guiding Example

Cake is often a desired dessert for receiving guests or special occasions and is generally easy to make. There are countless varieties of cakes but they tend to follow the same basic steps: Mixing, baking, and decoration. The first step, mixing, is responsible of mixing the cake’s ingredients in order to make cake batter. Basic ingredients used in this stage are usually butter, flour, sugar, and eggs. After the cook mix the ingredients for a period of time, generally until the cake batter acquires a uniform color, the batter goes to a cake pan in order to undergo the next stage: baking.

The baking stage takes the cake to the oven to bake the batter. Generally the baking process takes around 30 to 45 minutes, depending on the oven. After baking is complete, the cake must be left untouched to cool down and to be removed from the cake pan. When it is removed from the pan, the cake is ready for the last stage, which consists on decorating it with icing or any other eligible ingredients. Finally, when the decoration is over, the cake is ready to be served as dessert for house guests.

## **Open Provenance Model**

The OPM emerged as a result of the *Provenance Challenges* proposed in the context of IPAW. The *Provenance Challenges* came in four editions, one for each year from 2006 to 2010, except 2008, and OPM resulted from the first two challenges and was used on the third challenge:

*1st Challenge*: Aimed at providing a forum for the community to understand the capabilities of different provenance systems and express their provenance representations.

*2nd Challenge*: Aimed at establishing interoperability between systems through exchange of provenance information.

*3rd Challenge*: Evaluate the OPM in a practical setting, from an interoperability perspective.

The OPM is a provenance model designed to allow provenance information to be exchanged among systems using a common provenance model. It allows developers to build and share tools for such provenance model, while supports a digital representation of provenance for anything, whether it was digitally generated or not. Lastly, OPM defines a set of rules to identify valid inferences that can be done on provenance representations (MOREAU *et al.*, 2007).

OPM assumes that the provenance of objects is represented by a causality graph, which is a directed acyclic graph with annotations to capture more detailed information about the objects’ execution. According to MOREAU *et al.* (2007), a provenance graph is “*a record of a past or current execution, and not a description of something that could happen in the future*”. Provenance, according to OPM, is composed of three types of elements (artifacts, processes, and agents) connected by causal relationships. The following sections describe in details those aspects as well as the possibility of inferring provenance statements to simplify the provenance graph, if necessary.

### Types and Relations

The causality graph used by the OPM is composed of vertices, which can represent *Artifacts*, *Processes*, and *Agents*, and edges that represent causal relationships between vertices. The definitions for all three vertex types are described in the following: *“****Artifacts*** *are immutable pieces of state that can represent a physical object or its digital representation in a computer system.* ***Processes*** *are actions or a sequence of actions performed or caused by artifacts resulting in new artifacts.”* Lastly, *“****Agents*** *are contextual entities acting as a catalyst of a process that can enable, facilitate, control, or affect its execution”* (MOREAU *et al.*, 2007)*.*

The edges of the provenance graph represent causal relationships between its source and destination. The edges can belong to one of the five categories defined by MOREAU *et al.* (2007) and described at . Using the OPM notation, it is possible to generate a provenance graph for the cake’s baking process, which is illustrated at Figure 2. In the cake example, the ingredients, the oven, and the cake are *artifacts*. The mixing, baking, and decorating stages are *processes*. Lastly, the *agent* is the cooker.

# 

Figure 1: Edges in OPM. Adapted from MOREAU *et al.*(2007).

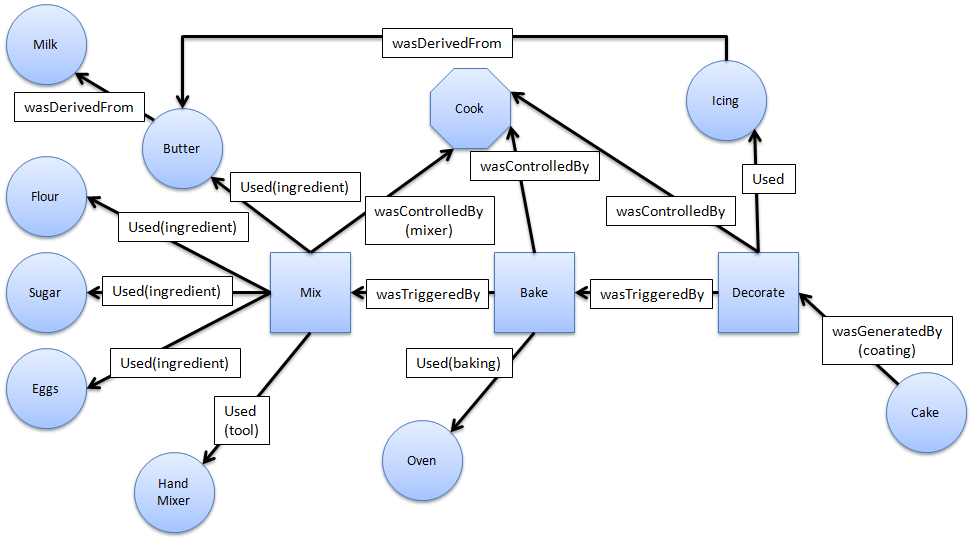


Figure 2: A simplified cake’s provenance graph.

A *used* edge that connects a *process* with an *artifact* indicates that the *process* required the *artifact* in order to complete its execution. If several *artifacts* are connected to the same *process*, then all of them were required for the *process* execution. It is also possible to specify each *artifact* role in the *process* execution by using the (*R)* field in the edge, shown in , which represents a role. Roles are not defined in OPM but are instead defined by the application domain. Using the cake example, the mix *process* used butter, flour, sugar, eggs, and a hand mixer. Their roles in the mix *process* are ingredient (butter, flour, sugar, and eggs) and tool (hand mixer).

The edge *wasGeneratedBy* connecting an *artifact* with a *process* indicates that the *artifact* was only generated after the *process* started its execution. If several *processes* are connected with an *artifact*, then all these *process* must have started for the *artifact* to be generated. Similar to *used* edge, it is possible to specify the *process’s* role in the generation of the *artifact*. In the cake example, the cake was generated by the *process* decorate with the role of coating the cake with icing to be a more appealing dessert for guests, instead of a simple cake.

A *wasControlledBy* edge connecting a *process* to an *agent* indicates that the *agent* controlled the *process’s* start and ending. If the *process* was controlled by multiple *agents*, then their roles can also be represented in the in the relationship. Note however that this type of dependency represents a control relationship and not a data derivation. In the cake example, the cook control the executions of the stages mix, bake, and decorate. He decides how long is the mixing, how long it takes to bake the cake, and how to decorate it.

The edge *wasTriggeredBy* connecting a *process* with another *process* indicates that the *process* was only able to complete after the other *process* started. This edge also allows for a *process* to have an oriented view of past executions. Using the cake example, the *process* bake was only able to start after the *process* mix had started and, in this case, finished as well.

Lastly, the edge *wasDerivedFrom* connecting an *artifact* with another *artifact* indicates that the *artifact* had to be generated for *artifact* to be generated. Using the cake example, the butter is made by churning milk. In other worlds, milk is necessary for the creation of butter, this way, butter was derived from the milk.

Moreover, OPMallows causality graphs to be used with time information. In OPM, time can be used to validate causality claims because if the same time clock is used to measure the passage of time for the effect and cause, then the effect will always come after the cause. In addition, time can be associated to *instantaneous occurrences* in a *process*. There are four types of occurrences (MOREAU *et al.*, 2007): *creation,* *use*, *starting* and *ending*. The first two are used for *artifacts* while the other two for *processes*. Because time is measured by someone, the model allows for a margin of error when representing time. For example, an *artifact* was used no earlier than time t1 and no later than time t2. This rationale is analogous for *processes*.

indicates that time information can be expressed in the model by using labels in the relationships. For *used* and *was generated by* edges, one timestamp can be used to express when the event happened. For *was controlled by* edge two timestamps mark when the process started and terminated. For *was derived from* and *was triggered by* edges, one timestamp to indicate when the *artifact* was generated. Despite using timestamp, the time of occurrence itself is not enough to imply causality. The fact that *process* P1 happened before P2 is not enough information to infer that P1 is the cause for P2 execution.

### Inference

The OPMalso has defined the notion of a graph based on a set of syntactic rules and topological constraints (MOREAU *et al.*, 2007). The provenance graph captures causal dependencies that can be summarized by inferring facts, using transitive rules. Because of this, the OPMdefined a set of completion rules and inferences that can be used in the graph to improve understanding by omitting unnecessary steps. There are three completion rules: *artifact introduction*, *artifact* *elimination*, *process introduction*, and *process elimination*.

For *artifact introduction*, Figure 3 illustrates the transformation of the *wasTriggeredBy* edge to *wasGeneratedBy* and *used* edges, introducing an *artifact* (cake batter). This is possible because the *wasTriggeredBy* edge is a composition of both edges. However, the identity of the *artifact* cannot be specified due to the lack of information, unless if the same *artifact* was previously present in the graph and removed by inference or specified somewhere else. Analogous, in the *artifact elimination* it is possible to transform the *used* and *wasGeneratedBy* edges to *wasTriggeredBy* edge by hiding the *artifact* (cake butter). This is possible because *wasTriggeredBy* edge is a composition of both edges.



Figure 3: Artifact introduction and elimination using the cake example.

Another rule is the *process introduction* illustrated by Figure 4, which is similar to *artifact introduction* but with a *process* instead of an *artifact*. While the *wasTriggeredBy* edge hides the presence of an *artifact*, the *wasDerivedFrom* edge hides the presence of an intermediary *process*. However, the *process introduction* rule can only be applied if butter is actually dependent of milk. This is the case, since in order to produce butter, it is necessary to churn the milk. So it is possible to add the *churning* *process* between milk and butter. The last rule, *process elimination*, is analogous to *artifact introducion*.

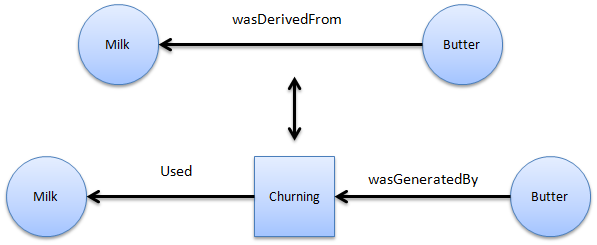


Figure 4: Process introduction and elimination in the cake example.

In a provenance graph, the causes of an *artifact* or a *process* might be traced to indirect causes from other *processes* and can involve multiple transitions between the cause and the perceived effect. For this purpose, OPM offers a set of relationships necessary for inferences: the *Multi-step edges* (MOREAU *et al.*, 2007). The multi-step edges can be composed of four types: *wasDerivedFrom*\*, *used*\*, *wasGeneratedBy*\*, and *wasTriggeredBy*\*. Unlike their normal counterparts, these edges represent that multiple steps were necessary to be taken to have the same meaning as their normal counterparts.

Using the cake example, it is possible to make inferences by using multi-step edges, which are represented as dashed edges in Figure 5. The edge *wasDerivedFrom*\* says that the *artifact* icing was indirectly derived from milk (using a multi-step edge), because the butter, which the icing is derived from, is in turn derived from milk. This states that the *artifact* milk had an influence over the *artifact* icing. Another possible multi-step is the *wasGeneratedBy*\* connecting *artifact* cake to *process* mix. This is possible because the *artifact* cake was generated by *processes* decorate, bake, and mix. Moreover, *process* decorate *wasTriggeredBy*\* *process* mix, since *process* decorated could only have started after *process* mix had started, and ended, in this example. Lastly, *process* bake *used*\* *artifact* eggs because eggs was necessary for the *process* mix, which in turn was necessary for *process* bake to start. As can be seen from these examples, multi-step edges can be used to infer single-step edges by eliminating *artifacts* and *processes* that occur in a chain of events.

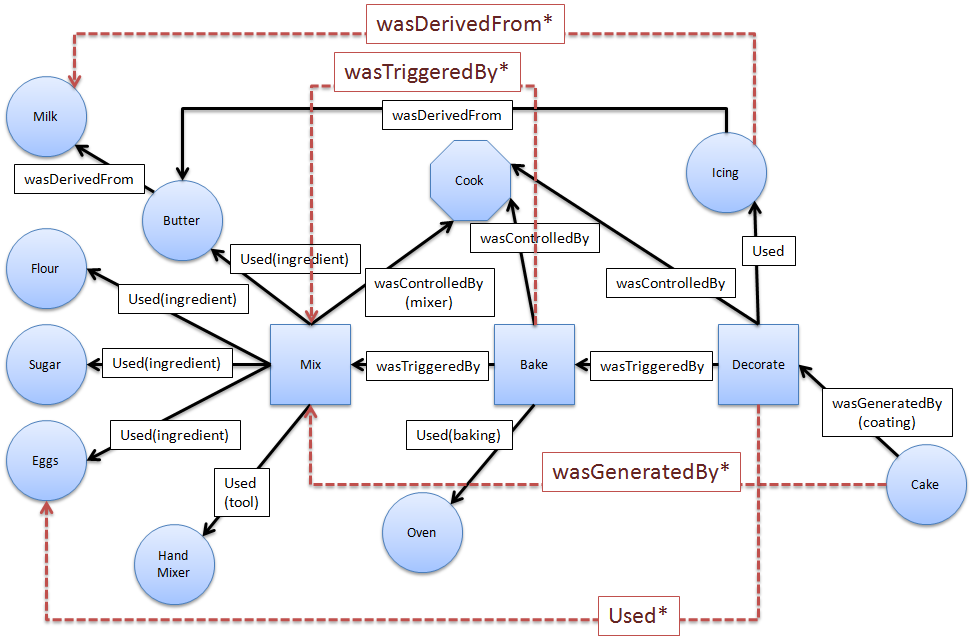


Figure 5: Multi-step edges in the cake example.

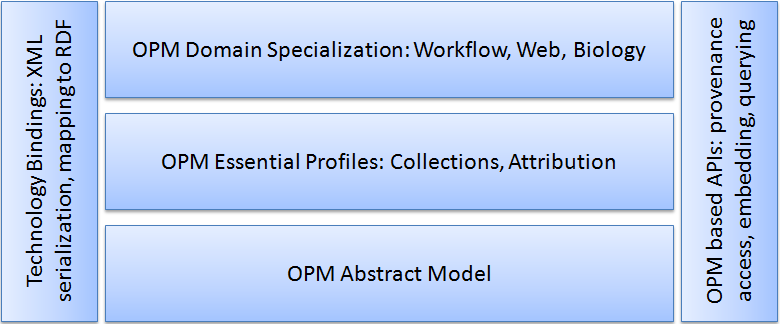


Figure 6: OPM’s Layered Architecture. Adapted from MOREAU *et al* (2007)

Lastly, the OPM has a modular design as illustrated by Figure 6. However, specifications for all layers in the design have not been produced, possibly because the development team started working on another provenance model: PROV. Nevertheless, at the bottom layer is located the abstract model (MOREAU *et al.*, 2007). On the left hand side, is located a serialization to *xml*, defined by OPMX (The Open Provenance Model XML Schema) (MOREAU; GROTH; *et al.*, 2010), a mapping to RDF with OPMV (The Open Provenance Model Vocabulary) (ZHAO, 2010) and OPMO (The Open Provenance Model OWL Ontology) (MOREAU; DING; *et al.*, 2010). Those are the only specifications produced, along with the *Open Provenance Model Java Library* (MOREAU, 2010b), and a JAXB-generated Java Library used by *OPM Toolbox* (MOREAU, 2010a) for creating a Java representation of OPM graphs and serializing them to or from a *XML* file. With the development of PROV, these other OPM specifications (Essential Profiles, Domain Specialization, and APIs) were left unfinished.

## **PROV**

PROV is a family of specifications proposed by W3C group to express provenance of digital objects, containing descriptions of the entities and activities involved in producing and delivering an object. In PROV, “*provenance is information about entities, activities, and people involved in producing a piece of data which can be used to form assessments about its quality, reliability or trustworthiness*” (NIES *et al.*, 2010). The goal of this provenance model is similar to OPM’s, allowing provenance information to be exchanged among systems using a common provenance model, while also enabling the provenance representation by a provenance graph (GROTH; MOREAU, 2010)..

The discussion group that developed PROV was officially launched concurrently to the forth *Provenance Challenge* (MILES *et al.*, 2010). The specifications that make up the PROV provenance model were divided into several documents that details different aspects of it: PROV *Overview (PROV-OVERVIEW)* (GROTH; MOREAU, 2010), PROV *Primer (PROV-PRIMER)* (GIL; MILES, 2010), Prov *Ontology (PROV-O)* (LEBO; SAHOO; MCGUINESS, 2010), PROV *Data Model (PROV-DM)* (MOREAU; MISSIER, 2010a), PROV *Constraints (PROV-CONSTRAINTS)* (NIES *et al.*, 2010), PROV *Notation (PROV-N)* (MOREAU; MISSIER, 2010b), PROV XML (PROV-XML) (HUA *et al.*, 2010), PROV *Dublin Core Mapping (PROV-DC)* (GARIJO *et al.*, 2010), PROV *Links* (MOREAU; LEBO, 2010), PROV *Access and Query (PROV-AQ)* (WEITZNER *et al.*, 2008), PROV *Dictionary (PROV-DICTIONARY)* (MISSIER *et al.*, 2010), PROV *Semantics (PROV-SEM)* (CHENEY, 2010), and PROV *Implementations* (GROTH *et al.*, 2012) .

Figure 7 illustrates the organization of PROV. At its core, is the conceptual data model that defines the common vocabulary used to describe provenance. Inside the data model, there is a set of constraints defined to aid developers in creating provenance programs to validate provenance statements. In order to support the interchange of provenance, PROV defined protocols to locate, access, and connect sets of provenance in order to aid in their interoperability.

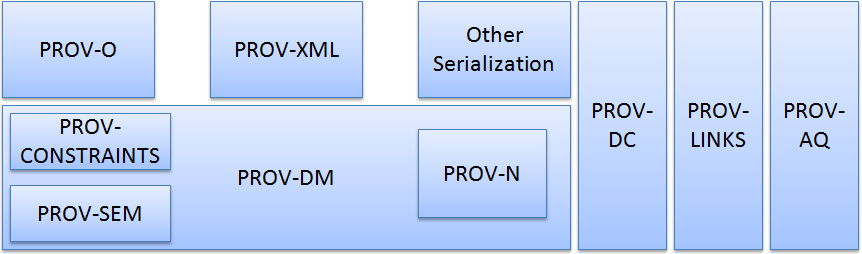


Figure 7: PROV organization. Adapted from GROTH and MOREAU (2010)

Provenance can be used for many purposes, from understanding how the data was collected in order to use it, to determine the object’s ownership, and decide if the information is trustworthy. It can also be used to check if the steps used in the process to obtain the result is compatible with the requirements. Lastly, it can be used to show the necessary steps to reproduce something. The PROV model specification accommodates these usages of provenance and provides three different ways to capture information according to the user’s perspective of provenance (GIL; MILES, 2010): agent-centered, object-centered, and process-centered.

The agent-centered approach focus on describing entities involved in the generation or manipulation of the information, while process-centered focus on capturing actions and steps used to generate the information. Lastly, the object-centered approach traces the origins of a document or artifact to other artifacts.

### Types and RELATIONS

PROV also uses a graph, similar to the provenance graph from OPM, to represent the provenance information. This graph is also characterized by having edges representing relationships between vertices and three types of vertices (GIL; MILES, 2010): *Entities*, *Activities*, and *Agents*.

Similarly to *artifacts* from OPM, *entities* represent physical or digital objects like a document, the web, or material objects. *Activities*, which are similar to *processes* in OPM, are actions taken to change or interact with *entities* or *agents*. Lastly, an *agent* is a person, software, organization, or *entities* that have responsibilities. This responsibility is a link to an *activity*. Several *agents* can have responsibilities over the same *activity* and a single *agent* can have responsibilities over several *activities*. *Agents* can also act on behalf of other *agents*, representing their interests when they are unavailable. These causal relations are some of the possible relationships available in PROV and are represented by edges in the provenance graph, similarly to OPM. Figure 8 illustrates those vertex types in the graph along with some possible relations between them. These relations, as well as other possible relations, are defined in the following paragraphs according to PROV-DM (MOREAU; MISSIER, 2010a).

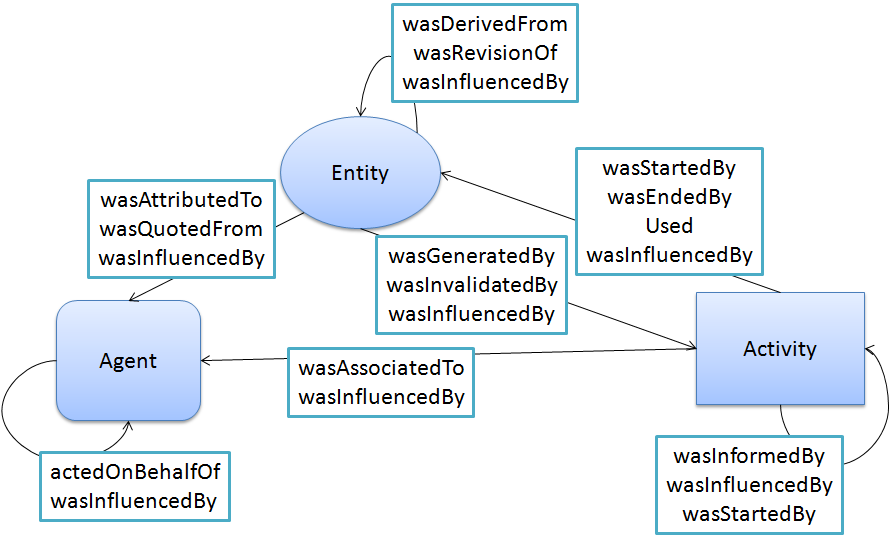


Figure 8: PROV Entities and possible relations. Adapted from GIL and MILES (2010)

Just like in OPM, the relationship denoted by a *used* edge from an *activity* to an *entity* indicates that the *activity* needed the *entity* for its operation. The relationship denoted by *wasStartedBy* edge from an *activity* to an *entity*, or between *activities*, indicates that the *entity* was the trigger that started the *activity*, similar to *wasTriggeredBy* from OPM. Similar to *wasStartedBy*, the *wasEndedBy* relationship indicates when the *activity* ended its execution by an *entity* or another *activity*.

The relationships *wasGeneratedBy* and *wasDerivredFrom* are also similar to their OPM’s counterparts. The *wasInvalidadedBy* relationship between *entities* indicates that the *entity* is no longer available for usage. For example, in the cake scenario if the butter was home-made and there was only enough milk to make it for the cake and not the icing, then the milk was invalidated by the butter, since there is no more milk left for making the icing. The *wasInformedBy* edge indicates an exchange of information using an unknown *entity* between *activities*. Using the cake example but in a scenario where the process is fully automated, this edge could be used for another *activity* to inform the bake *activity* how long the cake batter should bake to not ruin the cake. The relationship *wasAttributedTo* from an *entity* to an *agent* indicates that the *entity* is now assigned to that *agent*, while *wasAssociatedTo* from an *activity* to an *agent* indicates that the *agent* had a responsibility or a role in the *activity*.

The relationship between *agents* denoted by *actedOnBehalfOf* indicates that an *agent* assigned authority to another *agent* to execute *activities* on his behalf. The edge *wasRevisionOf* between *entities* indicates that the *entity* is a previous version of another *entity*. The edge *wasQuotedFrom* indicates that an *entity* was used by an *agent* and that agent was not its original owner. The *entity* must be a text or image taken from somewhere else, just like quoting what someone wrote. Lastly, the *wasInfluencedBy* edge indicates that the *entity, activity*, or *agent* affected another *entity, activity*, or *agent* by influencing it.

### Further Notations

Besides the relations mentioned in the previous subsection, the PROV model has support for a few more: specialization, alternate, and the possibility of extending existing structures. These extended structures are defined by a variety of mechanisms: subtyping, expanded relations, new relations, and optional identifiers.

According to MOREAU and MISSIER (2010a), an specialization, denoted by a *specializationOf* edge from an *entity* to another *entity* indicates that the first *entity* can do everything the second *entity* can, while also having more functions. Meanwhile, the *alternateOf* edge between *entities* indicates that both *entities* have the same functions and characteristics. Specialization and alternate are not considered as influences in PROV. Subtyping is a rule to create new edges based in existing relationships, *entities,* or *agents*. For example, the revision relationship is also a subtyping of the derivation relationship since a revision is a newer version based on the original document. PROV also supports the creation of totally new relationships without using any of the existing ones as a basis, which may be useful depending on the domain. For example, the relation *wasAttackedBy* is a relation between *agents* that indicates that one *agent* suffered an attack action from another *agent* in the game domain.

The relationships represented by edges between two vertices can also be expanded to add more details. For example, in a revision relationship, it is possible to add further details about the changes between versions, describing how the entity was altered to generate the newer version. Another example is plans used by agents. A plan, which in PROV can be represented as an entity, is a set of necessary steps to be taken in order to achieve the proposed goal. Figure 9 illustrates the usage of the *expanded relation* *plan* in the *hadPlan* edge connecting the edge *wasAssociatedWith*, from an *agent* (edith) and an *activity* (correct), with an *entity* (instructions).

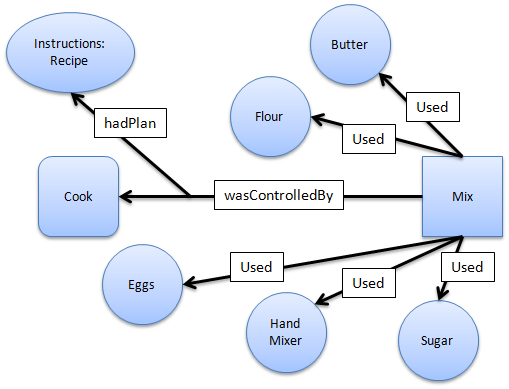


Figure 9: Using *Expanded Relations* in the cake example.

The PROV data model also has a set of pre-defined attributes to be used as optional identifiers to provide further details. There are five different types of attributes: *label*, *location*, *role*, *type*, and *value*. Labels are used to name an *agent*, *entity*, or *activity* for easier understanding, while locations are used to identify places that the agent, entity, or activity was located. Role, as previously mentioned, can be assigned to an *entity* or *agent* to provide an insight about their responsibilities during the execution of an *activity*. Type provides further information about the *agent*, *entity*, or *activity*. For example, *software agent* is a type of *agent* and chocolate cake is a type of cake. Finally, value is used to represent values associated with the entity, which can be a string, a number, or encoded data. Table 1 describes which constructs are allowed for each attribute and if there is any restriction of their values.

Table 1: PROV optional attributes. Adapted from MOREAU and MISSIER (2010a)

|  |  |  |
| --- | --- | --- |
| Attribute | Allowed in | Allowed Values |
| label | Any construct | *Value* of type *String* |
| location | *Entity*, *Activity*, *Agent*, *Usage*, *Generation*, *Invalidation*, *Start*, and *End* | *Any value* |
| role | *Usage*, *Generation*, *Invalidation*, *Association*, *Start* and *End* | *Any value* |
| type | Any construct | *Any value* |
| value | *Entity* | *Any value* |

### Time Information

The PROV model offers the ability to store information data from the time of origin due to the importance of temporal information in some scenarios. It is allowed to store date and time relating to *entities* or *activities*. For *entities*, it is allowed to store information regarding its generation or usage. As for *activities*, it is allowed to store information regarding when it started and ended its execution.

This information can be stored in tickets in the *activity* or in the relationships, as illustrated by Figure 10, showing the *startedAtTime* and *endedAtTime* in the *activity* bake, and the *generatedAtTime* in the relationship *wasGeneratedBy*. These tickets can also be used to store other information details, such as the usage of *Optional Identification* as mentioned in the previous subsection.

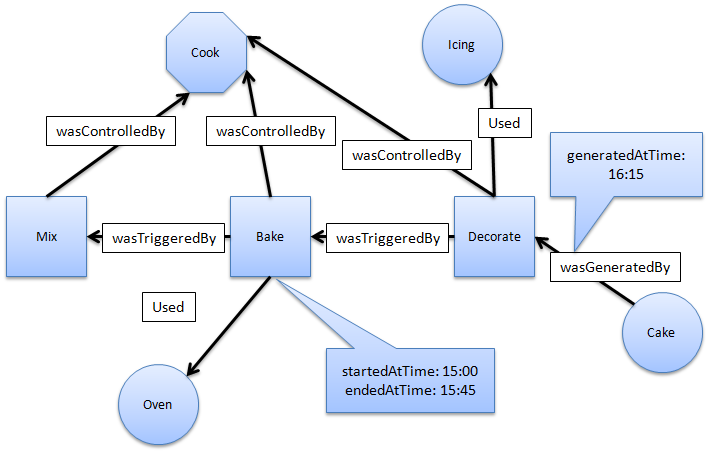


Figure 10: Time information using the cake example.

### Inference

Like OPM, the PROV model also supports the usage of inferences on provenance data to identify indirect effects or influences, while also preserving the meaning. A set of rules to validate a provenance inference in PROV is available at NEIS et al. (2010). An inference in PROV is a rule to add new statements or edges in order to simplify the information by skipping statements. It is also possible to replace statements or edges for equivalent ones to improve readability without changing the meaning of the statement (NIES *et al.*, 2010).

For example, it is possible to change a *wasDerivedFrom* relationship to a *wasRevisionOf* relationship if the meaning of the derivation in that situation is equal to a revision. This type of equivalence in PROV is called as *definitions*. Both definitions and inferences can be viewed as logical formulas (NIES *et al.*, 2010). However, while a *definition* can be used for both directions of an implication, this is not true for inferences. While looking at the *activities* necessary to generate a cake (*mix*, *bake*, *decoration*), it is possible to infer that the cake was generated by the *activity* mix. However, from the perspective of the *activity* *mix*, it is not possible to infer that *activities* *bake* and *decoration* were also involved.

## Comparison Between Models

In terms of key elements from both provenance models, it is possible to make a direct mapping between concepts by associating *artifacts*, *process*, and *agents* from OPM to *entities*, *activities*, and *agents* from PROV, respectively. Both models present ways of marking the passage of time (with timestamps) and execution, as well as providing rules for making inferences. However, PROV also provide explicit support for extending existing features, such as *subtyping*, expanding, and creating new relationships. Some relationships from both models are also compatible because, aside from having the same names, they also carry the same causal relationship between objects. These common relationships are: *used*, *wasGeneratedBy*, and *wasDerivedFrom*.

However, the relationship *wasControlledBy* from OPM doesn’t have one from PROV with the same name, but the relationship *wasAssociatedWith* from PROV has the same function, linking *activities* (*processes* in OPM) to *agents* in both models. Despite both relationships *wasTriggeredBy* from OPM and *wasInformedBy* from PROV being between two *processes* or *activities*, in PROV, they have different purposes. The relationship *wasInformedBy* aims to show that a particular *activity* reported something to the other, while *wasTriggedBy* from OPM indicates that a *process* has been initiated by another. However, there is a relationship in PROV (*wasStartedBy*) which equals to *wasTriggedBy* from OPM. Although the relationship *wasStartedBy* is more comprehensive as it can occur not only between two *activities*, but also between an *entity* and an *activity*.

Also, the PROV model has four relationships that were not found in OPM: the aforementioned *wasInformedBy*, *wasEndedBy*, *actedOnBehalfOf*, and *wasAttributedTo*. The relationships *actedOnBehalfOf* and *wasAttributedTo* are delegation and association of *agents* to *entities* and *activities*. These are important because PROV also aims at providing provenance information centered on *agents*, something that does not occur naturally in OPM. Lastly, the relationship *wasEndedBy* aims to represent the *activity’s* finalization. Table 2 illustrates the comparison of the existing relationships from both provenance models.

Table : OPM x PROV

|  |  |
| --- | --- |
| OPM | PROV |
| *used* | ***used*** |
| *wasGeneratedBy* | ***wasGeneratedBy*** |
| *wasControlledBy* | ***wasAssociatedWith*** |
| *wasDerivedFrom* | ***wasDerivedFrom*** |
| *wasTriggeredBy* | ***wasStartedBy*** |
|  | ***wasEndedBy*** |
|  | ***wasRevisionOf*** |
|  | ***wasAttributtedTo*** |
|  | ***wasInformedBy*** |
|  | ***actedOnBehalfOf*** |

From these relationships without direct equivalences, it is possible to observe differences between models. The OPM is simpler, and apparently aimed at controlling flows of execution taking particular attention on how a *process* was started by another. On the other hand, PROV appears to be more focused on issues of responsibility and historical data, having several relationships between *agents* and the other types (*entities* and *activities*), but also being more complete, having all relationships equivalent to the OPM. This may be due to the fact that the majority of OPM’s designers also participated in the creation of PROV.

## Final Considerations

This chapter presented the concepts of provenance in order to gather historical information about objects for further analysis. It was also presented both the existing provenance models (OPM and PROV) that can be used for provenance of digital information. Later, a comparison between models was made, pointing out their similarities. The incomplete OPM specifications may be a side effect from the fact that the same designers were involved in the creation of PROV, which occurred around the same year that OPM updates halted (at 2010). By analyzing both provenance models, there are three key points that inspired in the method used in this work for collecting gameplay data and representing the game flow in a provenance graph:

* Provenance of objects, which allows for a detailed study of an object’s life cycle.
* Provenance inferences, which allows making statements while, at the same time, hiding unimportant facts to reach conclusions about the object’s history.
* Provenance Graph, which allows for an analysis of the object’s interactions and influences from other *entities* throughout its life cycle by the means of a graph.

Thus, the next chapter presents the proposed approach to improve understanding of the game flow, providing insights in the story progression and how the outcome was influenced. The proposed approach is based on analyzing the game flow using a provenance graph and is called *Provenance in Games*.

# – Provenance in Games

## Introduction

## Final Considerations

# – Implementation

## Introduction

## Final Considerations

# – Evaluation

## Introduction

## Final Considerations

# – Conclusion

## Contributions

This paper introduces new perspectives on software engineering learning, leveraging the current state of the art, based on game, to a level where the game provenance can produce and consolidate knowledge. This knowledge can help on (1) confirming the hypotheses formulated by students, (2) supporting tutors for a better guidance, (3) motivating group dynamics around some case studies, and (4) extracting behavior patterns from individual sessions or groups of sessions.

The provenance visualization can occur both on-the-fly or in post-mortem sessions. It allows the discovery of issues that contributed to specific game flows and results achieved throughout the gaming session. This analysis can be used on games to improve understanding of the game flow and identifying actions that influenced the outcome, aiding the player to understand why they happened the way they did. It can also be used to analyze a game story development, how it was generated, and which events affected it.

Currently, we do not make inferences to the user, but let the user decide what he wants to infer. Studies in this area can be made in order to identify information that can be omitted from the user without affecting the overall analysis. Another interesting research is to automatically identify patterns in the game flow. Lastly, we are working on different graph visualization layouts and running experimental studies on the usage of provenance in educational games to evaluate the aspects of learnability.

## **Limitations**

## **Future Work**

In a future work, it is planned to to make *Proof Viewer* less context sensitive, allowing the user to customize filters (edge filters and attribute status visualization) without the need of hard coding it in the application. For example, allowing the user to provide a configuration file that specifies the type of each filter. Thus, this would make *Proof Viewer* compatible with other games or provenance applications without the need of tinkering with the source code.

Lastly, the *Proof Viewer’s* input file format. Currently, the *game flow log* is a simple tab separated value text file. However, there are plans to modify the structure to use some semi-structured format such as JSON or XML for greater compatibility with other applications. Thus, allowing *Proof Viewer* to be more accessible by other applications due to the *game flow log’s* XML format, making it easier for other games, or provenance applications, to use it.

### Scalability

Depending on the game style, a game session might take several hours to complete, or even days in case of RPGs. This makes the size of the provenance graph to be overwhelming to the user, even when making pre-filtering during the generation of the *game flow log*. One way to avoid such situations is to show the provenance graph with some filters selected instead of its full extension. For example, before showing the graph to the user, it is possible to use collapses to reduce the graph’s size. Combats can be identified and collapsed into a single vertex for each instance. Places visited in the game can also be collapsed into a single vertex, containing all interactions made in that location, even combats. It is also possible to have collapses inside collapses. In this case, a collapsed combat inside a collapsed area visited by the player may contain other actions aside from the combat, such as interactions with the ambient. This gives an impression of a map from the player’s journey, showing vertices for each location visited by the player, while allowing the player to expand only the situations he desires to analyze. It is similar to *google maps*, where it shows the entire world and allows the user to zoom into specific locations. However in this case, it shows instances of the journey taken by the player.

It is also possible to go beyond that. Instead of collapsing all combats and locations, filters can be used to decide which combats or locations were not relevant to the story, or had no noticeable impact in the player’s journey, while keeping important events visible to the player. This is possible because provenance is analyzed from the present to the past. This way, combats outcomes are known and can be used to decide if it they are relevant or not. If the player was victorious with minor challenge, did not suffer severe wounds, or barely used any resources at his disposal, then the entire combat can be simplified into just one vertex representing the combat with the enemy. However, if the combat was challenging or the player lost, it is interesting to display all actions in it for analysis, allowing the player to identify important facts that influenced the combat outcome.

# References

ABT, Clark C. *Serious Games*. 1. ed. Abt Books: University Press of America, 1987.

BAKER, Alex; NAVARRO, Emily; VAN DER HOEK, André. Problems and Programmers: An Educational Software Engineering Card Game. *International Conference on Software Engineering(ICSE)*, p. 614–621, 2003.

BOSE, Rajendra; FOSTER, Ian; MOREAU, Luc. Report on the International Provenance and Annotation Workshop: (IPAW’06) 3-5 May 2006, Chicago. *SIGMOD Rec.*, v. 35, n. 3, p. 51–53, set. 2006. Accessed: 2 abr. 2013.

BRISTOL, Edgar H. Pattern recognition: An alternative to parameter identification in adaptive control. *Automatica*, v. 13, n. 2, p. 197–202, mar. 1977. Accessed: 25 nov. 2012.

CHENEY, James. *Semantics of the PROV Data Model*. Available: <http://www.w3.org/TR/2013/WD-prov-sem-20130312/>. Accessed: 26 mar. 2013.

CHIALVO, D.R.; BAK, P. Learning from mistakes. *Neuroscience*, v. v. 90, n. 4, p. 1137–1148, jun. 1999. Accessed: 5 jul. 2012.

CIOS, Krzysztof; PEDRYCZ, Witold; SWINIARSKI, Roman W. *Data mining methods for knowledge discovery*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

CLARK, George. The organization of behavior: A neuropsychological theory. *The Journal of Comparative Neurology*, v. v. 93, n. 3, p. 459–460, 1950. Accessed: 5 jul. 2012.

DAVIDSON, Susan B.; FREIRE, Juliana. Provenance and scientific workflows: challenges and opportunities. SIGMOD ’08, 2008, New York, NY, USA. *Anais*... New York, NY, USA: ACM, 2008. p. 1345–1350. Available: <http://doi.acm.org/10.1145/1376616.1376772>. Accessed: 26 mar. 2013.

DIEHL, Stephan. *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*. [S.l.]: Springer, 2007.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, v. 17, n. 3, p. 37, 15 mar. 1996. Accessed: 25 nov. 2012.

FREIRE, J. *et al.* Provenance for Computational Tasks: A Survey. *Computing in Science Engineering*, v. 10, n. 3, p. 11–21, 2008a.

FREIRE, J. *et al.* Provenance for Computational Tasks: A Survey. *Computing in Science Engineering*, v. 10, n. 3, p. 11 –21, jun. 2008b.

GARIJO, Daniel *et al.* *Dublin Core to PROV Mapping*. Available: <http://www.w3.org/TR/2013/WD-prov-dc-20130312/>. Accessed: 26 mar. 2013.

GIL, Yolanda *et al.* Examining the Challenges of Scientific Workflows. *Computer*, v. 40, n. 12, p. 24–32, dez. 2007. Accessed: 2 abr. 2013.

GIL, Yolanda *et al.* *W3C Provenance Incubator Group*. Available: <http://www.w3.org/2005/Incubator/prov/wiki/Main\_Page>. Accessed: 22 mar. 2013.

GIL, Yolanda; MILES, Simon. *PROV Model Primer*. Available: <http://www.w3.org/TR/prov-primer/>. Accessed: 21 mar. 2013.

GROTH, Paul *et al.* *ProvImplementations*. Available: <http://www.w3.org/2011/prov/wiki/ProvImplementations>. Accessed: 26 mar. 2013.

GROTH, Paul; MOREAU, Luc. *PROV-Overview*. Available: <http://www.w3.org/TR/prov-overview/>. Accessed: 26 mar. 2013.

HAN, Jiawei; KAMBER, Micheline. *Data Mining: Concepts and Techniques*. [S.l.]: Morgan Kaufmann, 2006.

HUA, Hook *et al.* *PROV-XML: The PROV XML Schema*. Available: <http://www.w3.org/TR/prov-xml/>. Accessed: 26 mar. 2013.

JOSHUA O’MADADHAIN; DANYEL FISHER; TOM NELSON. *JUNG: Java Universal Network/Graph Framework*. sourceforge: Open-source, 2010. Available: <http://jung.sourceforge.net/>.

KOHWALTER, Troy; CLUA, Esteban; MURTA, Leonardo. Provenance in Games. In: 2012 XI BRAZILIAN SYMPOSIUM ON GAMES AND DIGITAL ENTERTAINMENT (SBGAMES), nov. 2012, Brasilia. *Anais*... Brasilia: In: XI SBGames, nov. 2012.

KOHWALTER, Troy; CLUA, Esteban; MURTA, Leonardo. SDM – An Educational Game for Software Engineering. In: 2011 X BRAZILIAN SYMPOSIUM ON GAMES AND DIGITAL ENTERTAINMENT (SBGAMES), nov. 2011, Salvador. *Anais*... Salvador: In: X SBGames, nov. 2011.

LEBO, Timothy; SAHOO, Satya; MCGUINESS, Deborah. *PROV-O: The PROV Ontology*. Available: <http://www.w3.org/TR/prov-o/>. Accessed: 21 mar. 2013.

LINDLEY, Craig A. Narrative, Game Play, and Alternative Time Structures for Virtual Environments. In: GÖBEL, STEFAN *et al.* (Org.). *Technologies for Interactive Digital Storytelling and Entertainment*. Lecture Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 2004. p. 183–194. Available: <http://link.springer.com/chapter/10.1007/978-3-540-27797-2\_25>. Accessed: 9 abr. 2013.

MISSIER, Paolo *et al.* *PROV Dictionary*. Available: <http://www.w3.org/TR/2013/WD-prov-dictionary-20130312/>. Accessed: 26 mar. 2013.

MOREAU, Luc *et al.* *IPAW*. Available: <http://www.ipaw.info/>. Accessed: 2 abr. 2013.

MOREAU, Luc; DING, Li; *et al.* *Open Provenance Model (OPM) OWL Specification*. Available: <http://openprovenance.org/model/opmo>. Accessed: 2 abr. 2013.

MOREAU, Luc; GROTH, Paul; *et al.* *Open Provenance Model (OPM) XML Schema Specification*. Available: <http://openprovenance.org/model/opmx>. Accessed: 2 abr. 2013.

MOREAU, Luc. *OPM Toolbox*. Available: <http://openprovenance.org/toolbox.html>. Accessed: 2 abr. 2013a.

MOREAU, Luc. *OPM4J: The Open Provenance Model Java Library*. Available: <http://openprovenance.org/java/site/1\_1\_8/apidocs/org/openprovenance/model/package-summary.html>. Accessed: 2 abr. 2013b.

MOREAU, Luc *et al.* The Open Provenance Model core specification (v1.1). *In: Future Generation Computer Systems*, v. 27, n. 6, p. 743–756, 2007. Accessed: 10 jun. 2012.

MOREAU, Luc; LEBO, Timothy. *Linking Across Provenance Bundles*. Available: <http://www.w3.org/TR/2013/WD-prov-links-20130312/>. Accessed: 26 mar. 2013.

MOREAU, Luc; MISSIER, Paolo. *PROV-DM: The PROV Data Model*. Available: <http://www.w3.org/TR/prov-dm/>.

MOREAU, Luc; MISSIER, Paolo. *PROV-N: The Provenance Notation*. Available: <http://www.w3.org/TR/prov-n/>. Accessed: 21 mar. 2013b.

NAVARRO, Emily. *Simse: a software engineering simulation environment for software process education*. 2006. California State University at Long Beach, Long Beach, CA, USA, 2006.

NIES, Tom De *et al.* *Constraints of the PROV Data Model*. Available: <http://www.w3.org/TR/prov-constraints/>. Accessed: 21 mar. 2013.

OXLAND, Kevin. *GAMEplay and Design*. [S.l.]: Addison-Wesley, 2004.

PREMIS WORKING GROUP. *Data Dictionary for Preservation Metadata*. Technical, no Final Report of the PREMIS Working Group. Preservation Metadata: Implementation Strategies (PREMIS), 2005. Available: <http://www.oclc.org/research/projects/ pmwg/premis-final.pdf>.

PRENSKY, Marc. Fun, Play and Games: What Makes Games Engaging. *Digital Game-Based Learning*, p. 1–31, 2001.

*Provenance Challenge WIKI*. Available: <http://twiki.ipaw.info/bin/view/Challenge/>. Accessed: 26 mar. 2013.

SIMMHAN, Yogesh L.; PLALE, Beth; GANNON, Dennis. A survey of data provenance in e-science. *SIGMOD Rec.*, v. 34, n. 3, p. 31–36, set. 2005. Accessed: 2 abr. 2013.

WEITZNER, Daniel J. *et al.* Information accountability. *Communications of the ACM*, v. 51, n. 6, p. 82–87, 1 jun. 2008. Accessed: 21 mar. 2013.

WITTEN, Ian H.; FRANK, Eibe. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. [S.l.]: Morgan Kaufmann, 2005.

ZHAO, Jun. *Open Provenance Model Vocabulary Specification*. Available: <http://open-biomed.sourceforge.net/opmv/ns.html>. Accessed: 2 abr. 2013.

# Glossary

"Elemento opcional, elaborado em ordem alfabética" (ABNT, 2005).

# Appendix A – Título do Apêndice

"Elemento opcional. O(s) apêndice(s) são identificados por letras maiúsculas consecutivas, travessão e pelos respectivos títulos. Excepcionalmente utilizam-se letras maiúsculas dobradas, na identificação, quando esgotadas as 23 letras do alfabeto" (ABNT, 2005).

# Annex A – Título do Anexo

"Elemento opcional. O(s) anexo(s) são identificados por letras maiúsculas consecutivas, travessão e pelos respectivos títulos. Excepcionalmente utilizam-se letras maiúsculas dobradas, na identificação dos anexos, quando esgotadas as 23 letras do alfabeto" (ABNT, 2005).

# Index

"Elemento opcional, elaborado conforme a ABNT NBR 6034" (ABNT, 2005).