Diagramas UML como herramienta para el diseño de provenance

CARLOS SÁENZ ADÁN

Índice

- Definición de *provenance*
- Ejemplos de *provenance*
- W3C PROV standard
- PROV-Templates
- Methodology overview
- De UML a PROV
  - UML Class Diagrams
  - UML Sequence Diagrams (revisited)
- Aplicación de la programación orientada a aspectos para la generación de *bindings.*
- Ejemplo

Índice

- **Definición de *provenance***
- Ejemplos de *provenance*
- W3C PROV standard
- PROV-Templates
- Methodology overview
- De UML a PROV
  - UML Class Diagrams
  - UML Sequence Diagrams (revisited)
- Aplicación de la programación orientada a aspectos para la generación de *bindings.*
- Ejemplo

Definición de *provenance*

The W3C (World Wide Web Consortium) Provenance Working Group's definition of provenance:

"Provenance is defined as a record that **describes** the people, institutions, entities, and activities **involved** in producing, influencing, or delivering a piece of data or a thing in the world"

"P*rovenance* se define como un registro que **describe** a las personas, instituciones, entidades y actividades **involucradas** con la producción, la influencia o la entrega de un dato o <<una cosa>> en el mundo"

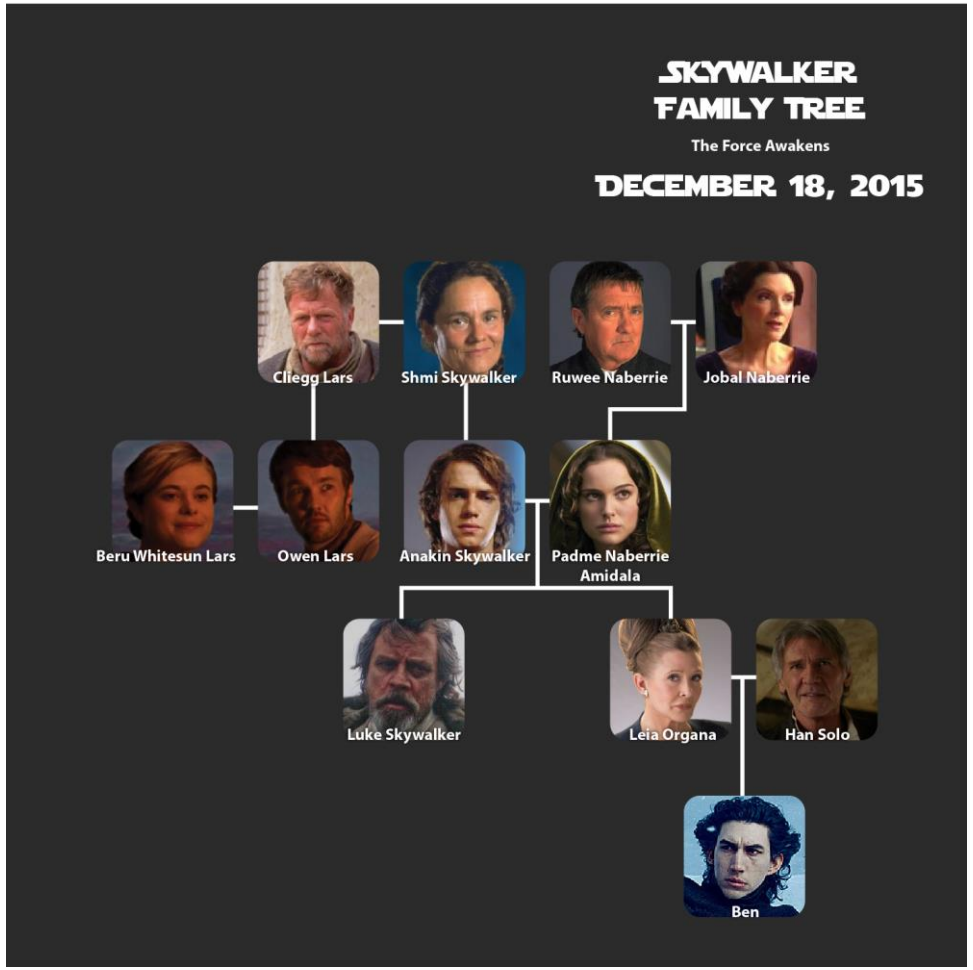
Índice

- Definición de *provenance*
- **Ejemplos de *provenance***
- W3C PROV standard
- PROV-Templates
- Methodology overview
- De UML a PROV
  - UML Sequence Diagrams (revisited)
  - UML Class Diagrams
- Aplicación de la programación orientada a aspectos para la generación de *bindings*.
- Ejemplo

Ejemplo de *provenance*

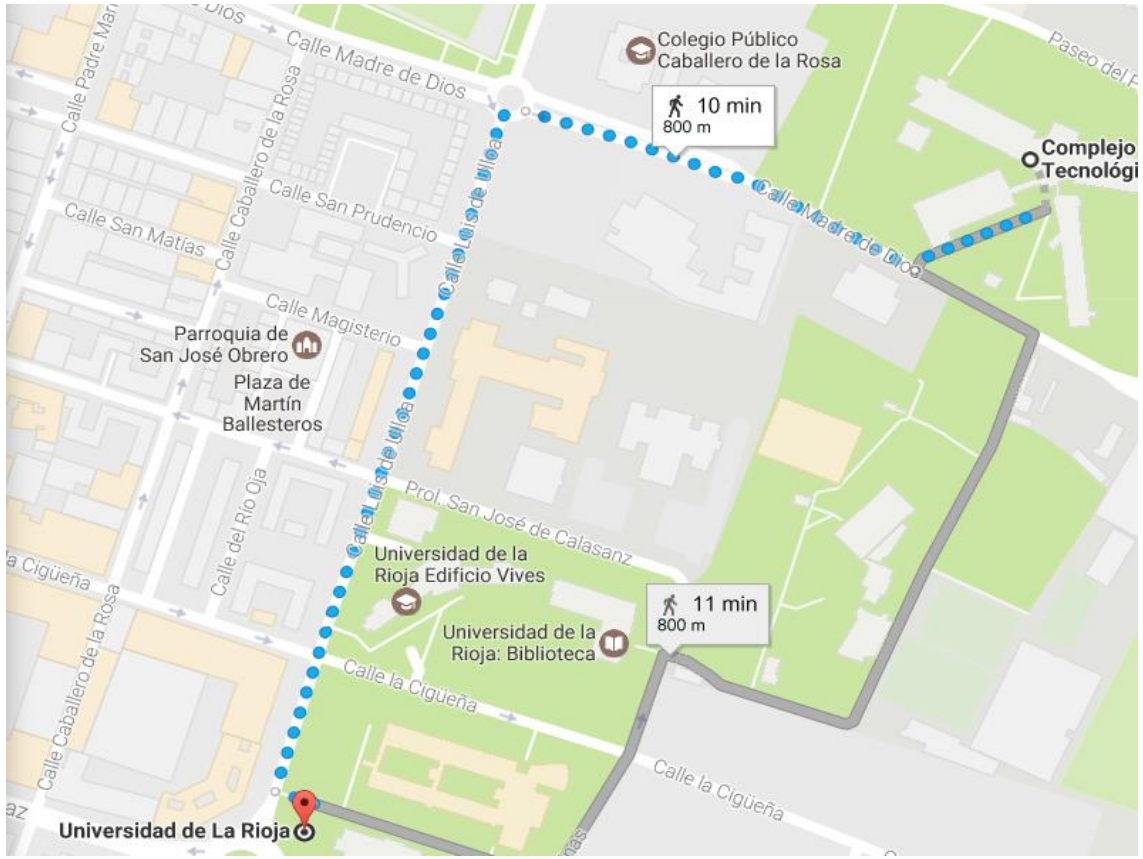
Hansel y Gretel

Ejemplo de *provenance*



Árbol genealógico

Ejemplo de *provenance*



Una ruta seguida

Ejemplo de *provenance*

```
Exception in thread "main" java.lang.NullPointerException
        at templatematching.TestMatchWebCamArea.<init>(TestMatchWebCamArea.java:90)
        at templatematching.TestMatchWebCamArea.main(TestMatchWebCamArea.java:132)
```

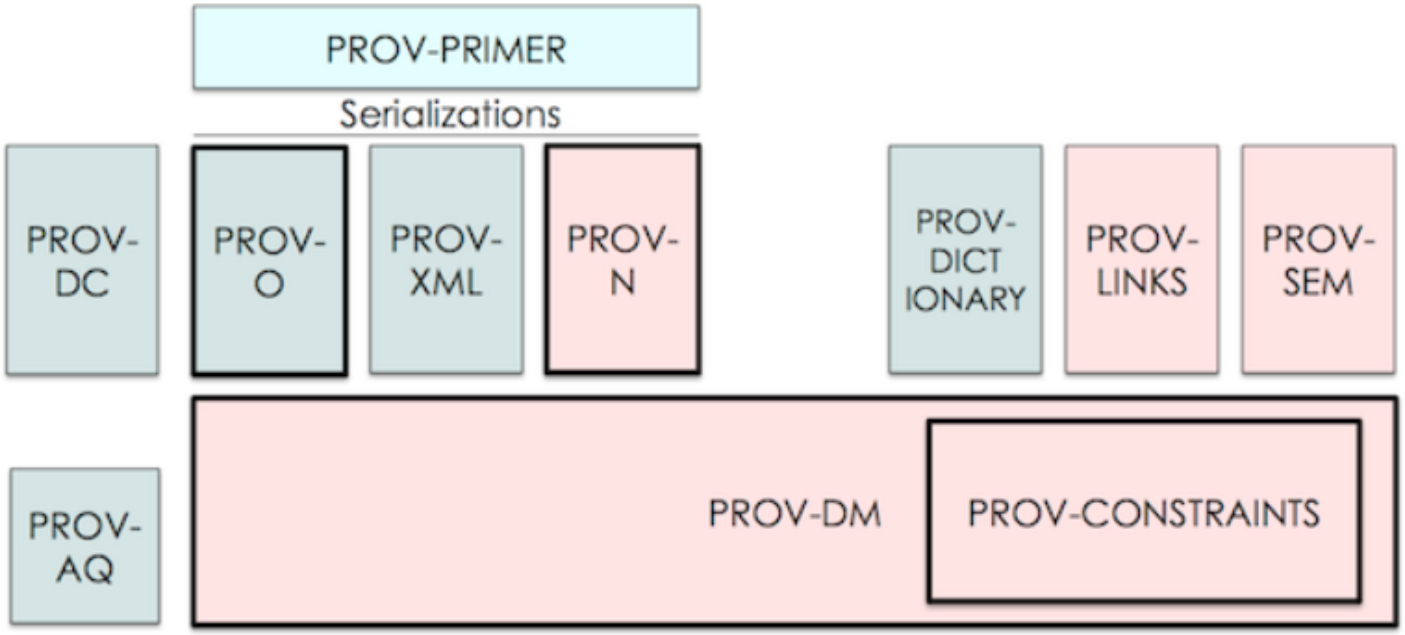
StackTrace

Índice

- Definición de *provenance*
- Ejemplos de *provenance*
- **W3C PROV standard**
- PROV-Templates
- Methodology overview
- De UML a PROV
  - UML Sequence Diagrams (revisited)
  - UML Class Diagrams
- Aplicación de la programación orientada a aspectos para la generación de *bindings*.
- Ejemplo

Provenance estándar. W3C PROV

Family of documents



Users. Quieren entender PROV y usar aplicaciones que soportan PROV

Developers. Quieren desarrollar aplicaciones que generan y consumen PROV

Advanced. Quieren crear validadores, nuevas formas de serialización, o sistemas avanzados de provenance.

https://www.w3.org/TR/prov-overview/

Provenance estándar. W3C PROV

Family of documents

PROV-OVERVIEW. Descripción general de la familia de documentos PROV.
PROV-PRIMER. Manual básico del modelo de datos PROV.
PROV-O. Ontología de PROV en OWL2. Permite el mapeo entre PROV y RDF.
PROV-DM. El modelo de datos PROV para provenance.
PROV-N. Notación PROV legible para las personas.
PROV-CONSTRAINTS. Conjunto de restricciones aplicables al modelo de datos PROV.
PROV-XML. XML Schema del modelo de datos PROV
PROV-AQ. Mecanismo para el acceso y consulta de provenance.
PROV-DICTIONARY. Añade un nuevo tipo de colección.
PROV-DC. Mapping entre PROV-O y Dublin Core Terms.
PROV-SEM. Especificación en términos de lógica de primer orden.
PROV-LINKS. Mecanismo para relacionar bundles.

Provenance estándar. W3C PROV

Family of documents

PROV-OVERVIEW. Descripción general de la familia de documentos PROV.
PROV-PRIMER. Manual básico del modelo de datos PROV.
PROV-O. Ontología de PROV en OWL2. Permite el mapeo entre PROV y RDF.
PROV-DM. El modelo de datos PROV para provenance.
PROV-N. Notación PROV legible para las personas.
PROV-CONSTRAINTS. Conjunto de restricciones aplicables al modelo de datos PROV.
PROV-XML. XML Schema del modelo de datos PROV
PROV-AQ. Mecanismo para el acceso y consulta de provenance.
PROV-DICTIONARY. Añade un nuevo tipo de colección.
PROV-DC. Mapping entre PROV-O y Dublin Core Terms.
PROV-SEM. Especificación en términos de lógica de primer orden.
PROV-LINKS. Mecanismo para relacionar bundles.

Provenance estándar. W3C PROV

Elementos

Entity: es algo físico, digital, conceptual o de otro tipo con algunas características fijadas. Puede ser real o imaginario.

Activity: algo que ocurre durante un periodo de tiempo y actúa sobre entidades, puede incluir el consumo, procesamiento, transformación, modificación, uso y generación de entidades.

Agent: algo que tiene algún tipo de responsabilidad sobre la ejecución de una actividad, la existencia de una entidad o la actividad de otro agente.

Provenance estándar. W3C PROV

Relaciones

		Object		
		Entity	Activity	Agent
Subject	Entity	• WasDerivedFrom • HadMember	• WasGeneratedBy	• WasAttributedTo
	Activity	• Used • WasStartedBy • WasEndedBy	• WasInformedBy	• WasAssociatedWith
	Agent			• ActedOnBehalfOf

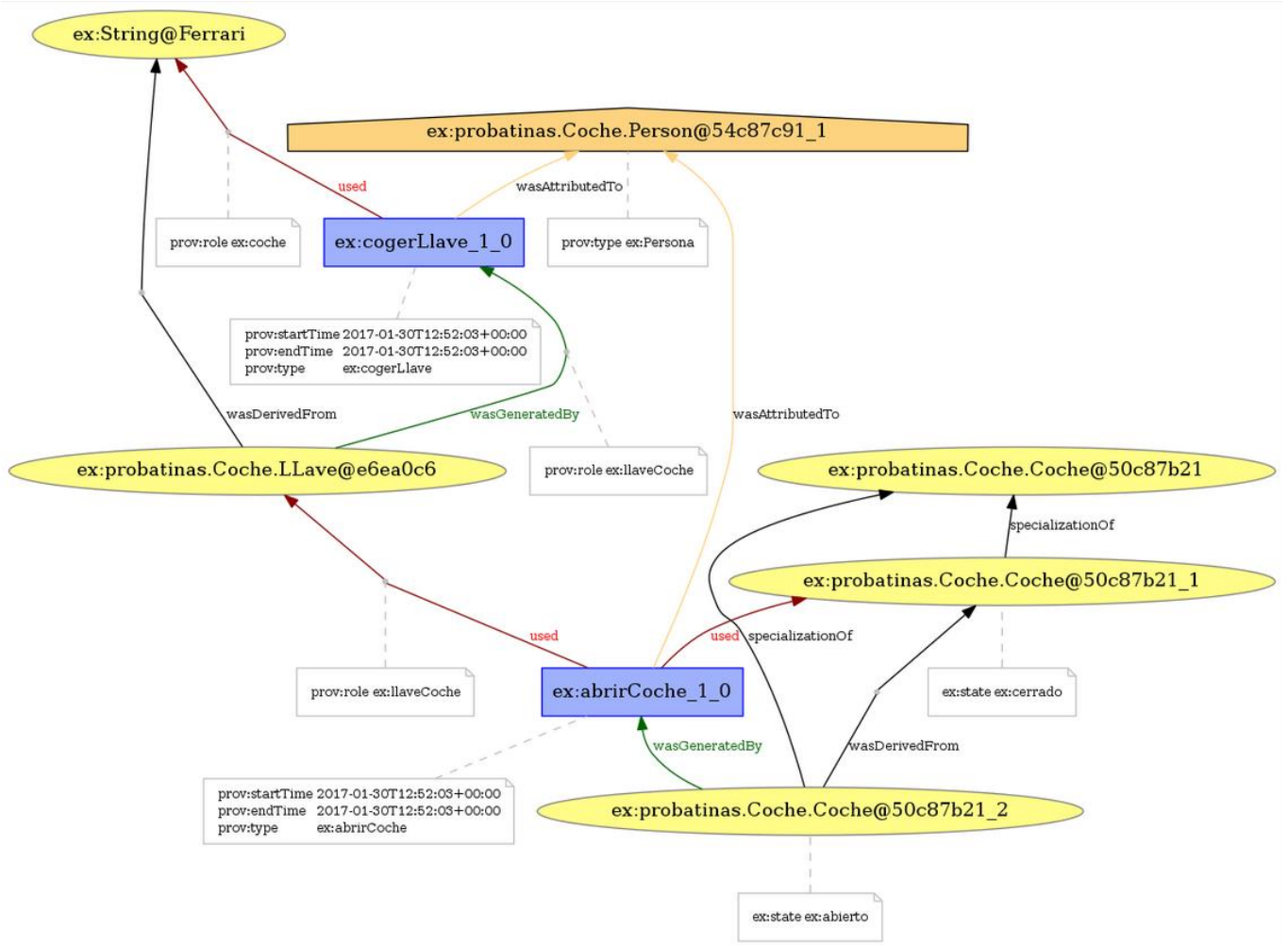
Provenance estándar. W3C PROV

Relaciones

		Object		
		Entity	Activity	Agent
Subject	Entity	• **WasDerivedFrom** • **HadMember**	• **WasGeneratedBy**	• **WasAttributedTo**
	Activity	• **Used** • **WasStartedBy** • WasEndedBy	• WasInformedBy	• **WasAssociatedWith**
	Agent			• ActedOnBehalfOf

Provenance estándar. W3C PROV

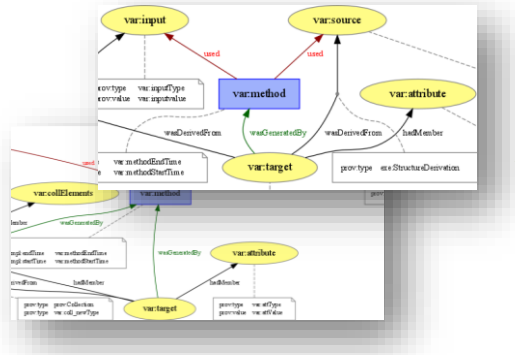
Relationships



Índice

- Definición de *provenance*
- Ejemplos de *provenance*
- W3C PROV standard
- **PROV-Templates**
- Methodology overview
- De UML a PROV
  - UML Sequence Diagrams (revisited)
  - UML Class Diagrams
- Aplicación de la programación orientada a aspectos para la generación de *bindings.*
- Ejemplo

PROV-Templates

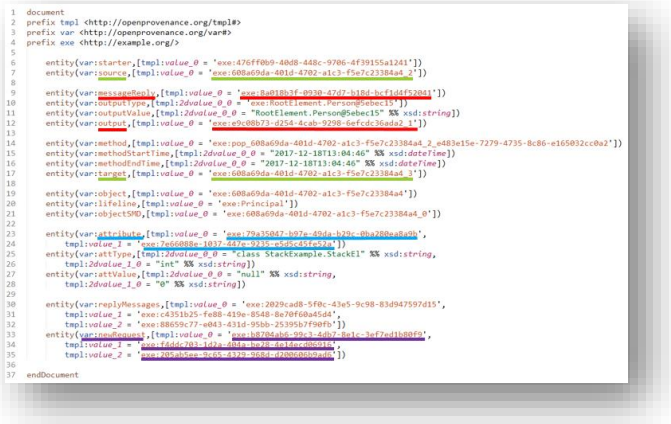


Usa

Expansión

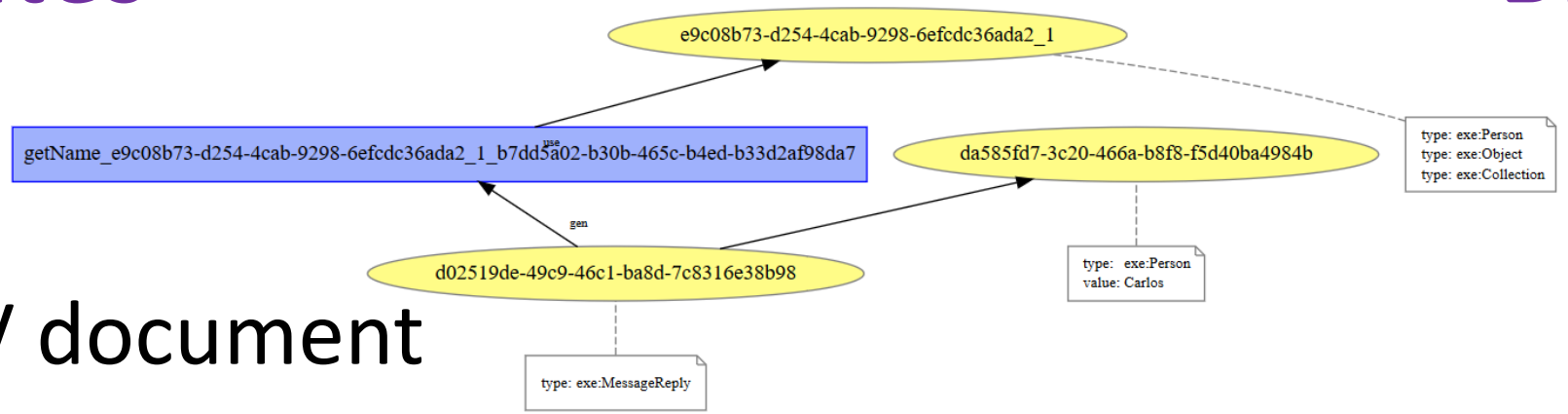
genera

Usa



Templates

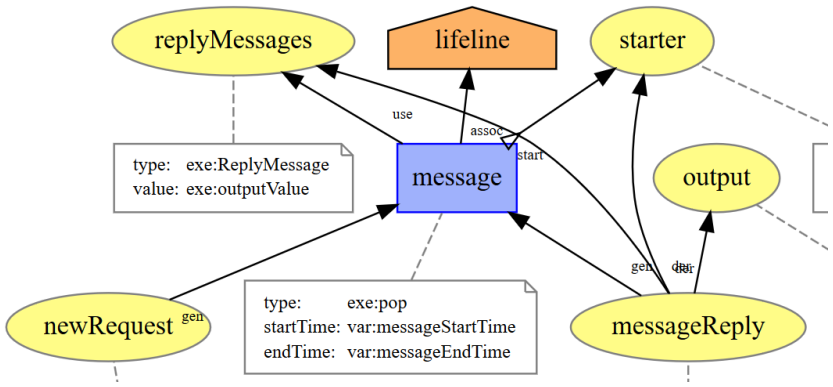
Bindings



PROV document

PROV-Templates

Templates



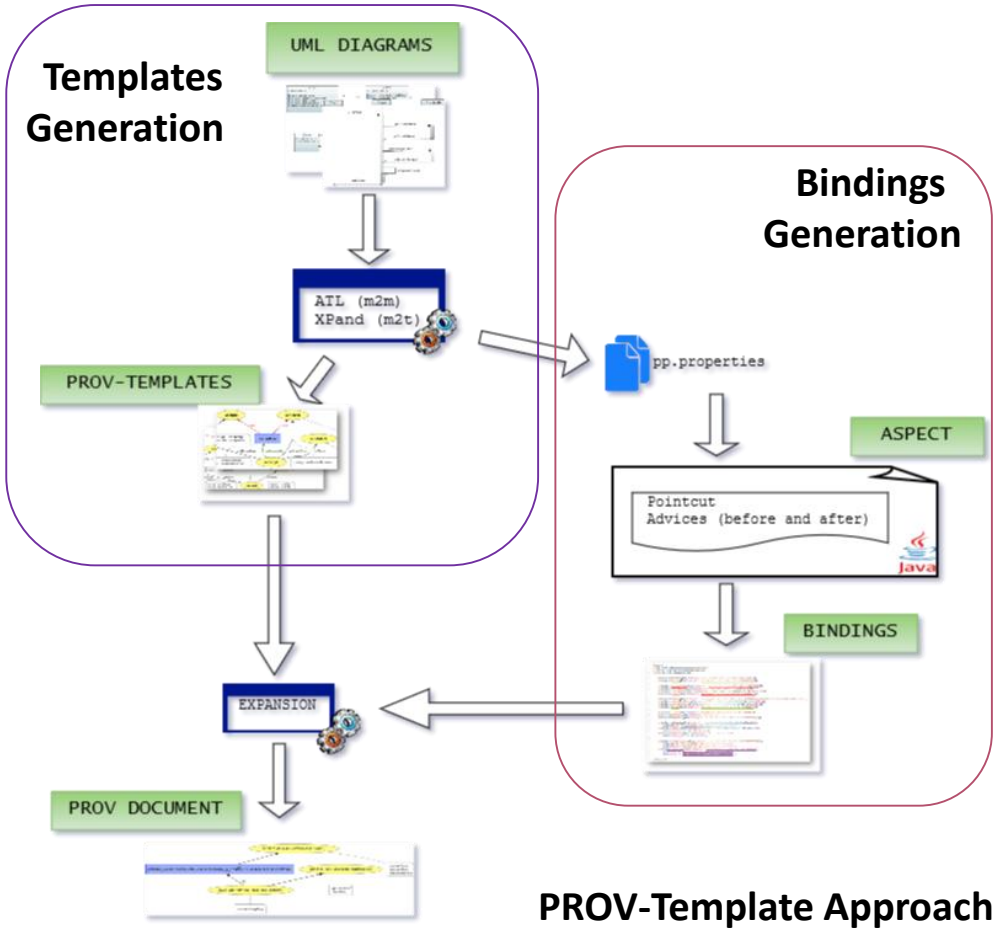
Bindings

```
document
prefix tmpl <http://openprovenance.org/tmpl#>
prefix var <http://openprovenance.org/var#>
prefix exe <http://example.org/>

    entity(var:starter,[tmpl:value_0 = 'exe:476ff0b9-40d8-448c-9706-4f39155a1241'])
    entity(var:source,[tmpl:value_0 = 'exe:608a69da-401d-4702-a1c3-f5e7c23384a4_2'])

    entity(var:messageReply,[tmpl:value_0 = 'exe:8a018b3f-0930-47d7-b18d-bcf1d4f52041'])
    entity(var:outputType,[tmpl:2dvalue_0_0 = 'exe:RootElement.Person@5ebec15'])
    entity(var:outputValue,[tmpl:2dvalue_0_0 = "RootElement.Person@5ebec15" %% xsd:string])
    entity(var:output,[tmpl:value_0 = 'exe:e9c08b73-d254-4cab-9298-6efcdc36ada2_1'])

    entity(var:method,[tmpl:value_0 = 'exe:pop_608a69da-401d-4702-a1c3-f5e7c23384a4_2_e483e15e-7279-4735-8c86-e165032cc0a2'])
    entity(var:methodStartTime,[tmpl:2dvalue_0_0 = "2017-12-18T13:04:46" %% xsd:dateTime])
    entity(var:methodEndTime,[tmpl:2dvalue_0_0 = "2017-12-18T13:04:46" %% xsd:dateTime])
    entity(var:target,[tmpl:value_0 = 'exe:608a69da-401d-4702-a1c3-f5e7c23384a4_3'])

    entity(var:object,[tmpl:value_0 = 'exe:608a69da-401d-4702-a1c3-f5e7c23384a4'])
    entity(var:lifeline,[tmpl:value_0 = 'exe:Principal'])
    entity(var:objectSMD,[tmpl:value_0 = 'exe:608a69da-401d-4702-a1c3-f5e7c23384a4_0'])

    entity(var:attribute,[tmpl:value_0 = 'exe:79a35047-b97e-49da-b29c-0ba280ea8a9b',
        tmpl:value_1 = 'exe:7e66088e-1037-447e-9235-e5d5c45fe52a'])
    entity(var:attType,[tmpl:2dvalue_0_0 = "class StackExample.StackEl" %% xsd:string,
        tmpl:2dvalue_1_0 = "int" %% xsd:string])
    entity(var:attValue,[tmpl:2dvalue_0_0 = "null" %% xsd:string,
        tmpl:2dvalue_1_0 = "0" %% xsd:string])

    entity(var:replyMessages,[tmpl:value_0 = 'exe:2029cad8-5f0c-43e5-9c98-83d947597d15',
        tmpl:value_1 = 'exe:c4351b25-fe88-419e-8548-8e70f60a45d4',
        tmpl:value_2 = 'exe:88659c77-e043-431d-95bb-25395b7f90fb'])
    entity(var:newRequest,[tmpl:value_0 = 'exe:b8704ab6-99c3-4db7-8e1c-3ef7ed1b80f9',
        tmpl:value_1 = 'exe:f4ddc703-1d2a-404a-be28-4e14ecd06916',
        tmpl:value_2 = 'exe:205ab5ee-9c65-4329-968d-d200606b9ad6'])

endDocument
```

Índice

- Definición de *provenance*
- Ejemplos de *provenance*
- W3C PROV standard
- **PROV-Templates**
- **Methodology overview**
- De UML a PROV
  - UML Class Diagrams
  - UML Sequence Diagrams (revisited)
- Aplicación de la programación orientada a aspectos para la generación de *bindings.*
- Ejemplo

Methodology overview

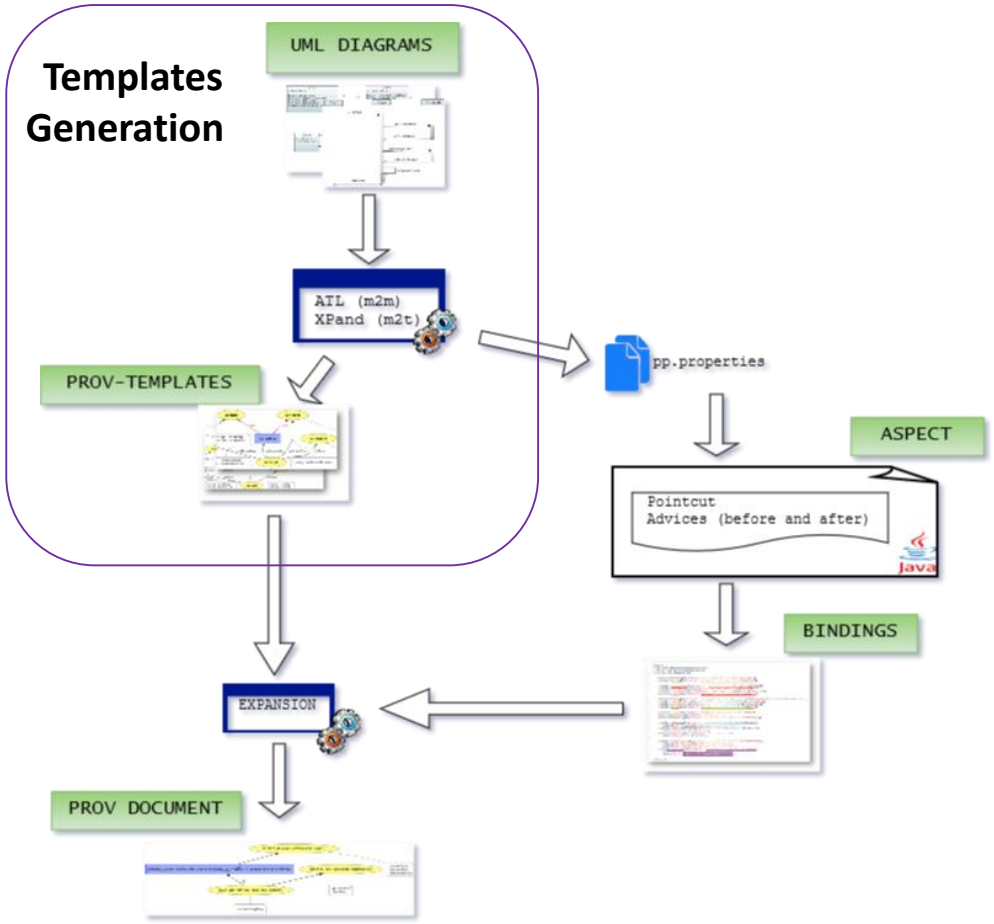


PROV-Template Approach

Índice

- Definición de *provenance*
- Ejemplos de *provenance*
- W3C PROV standard
- PROV-Templates
- Methodology overview
- **De UML a PROV**
  - UML Class Diagrams
  - UML Sequence Diagrams (revisited)
- Aplicación de la programación orientada a aspectos para la generación de *bindings.*
- Ejemplo

Methodology overview

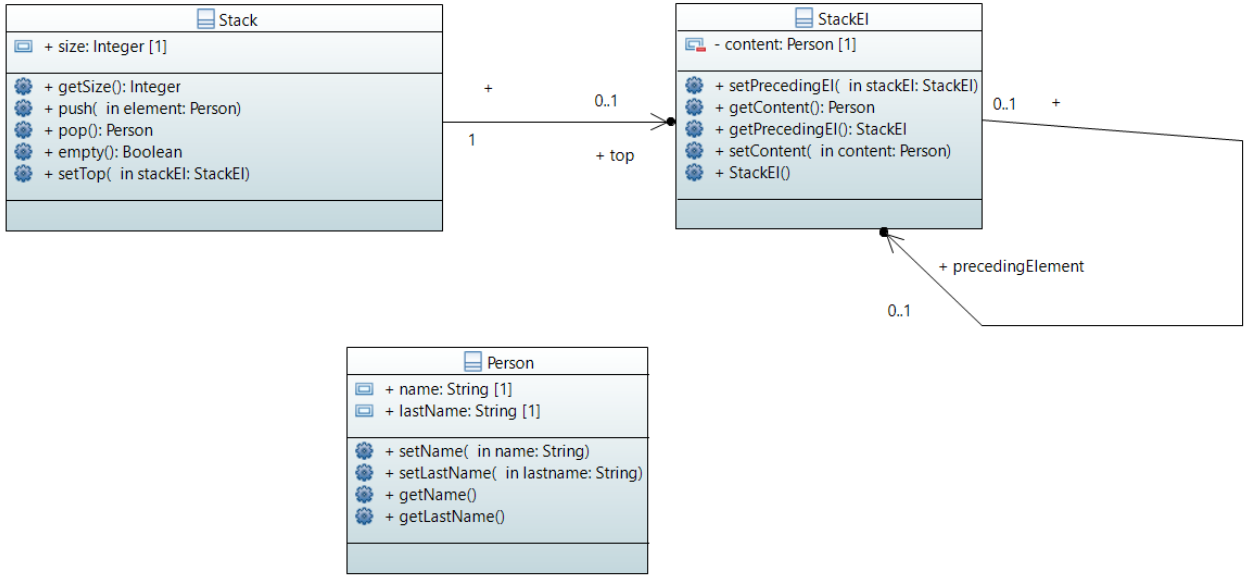


Índice

- Definición de *provenance*
- Ejemplos de *provenance*
- W3C PROV standard
- PROV-Templates
- Methodology overview
- **De UML a PROV**
  - **UML Class Diagrams**
  - UML Sequence Diagrams (revisited)
- Aplicación de la programación orientada a aspectos para la generación de *bindings.*
- Ejemplo

UML Class Diagrams

Definition

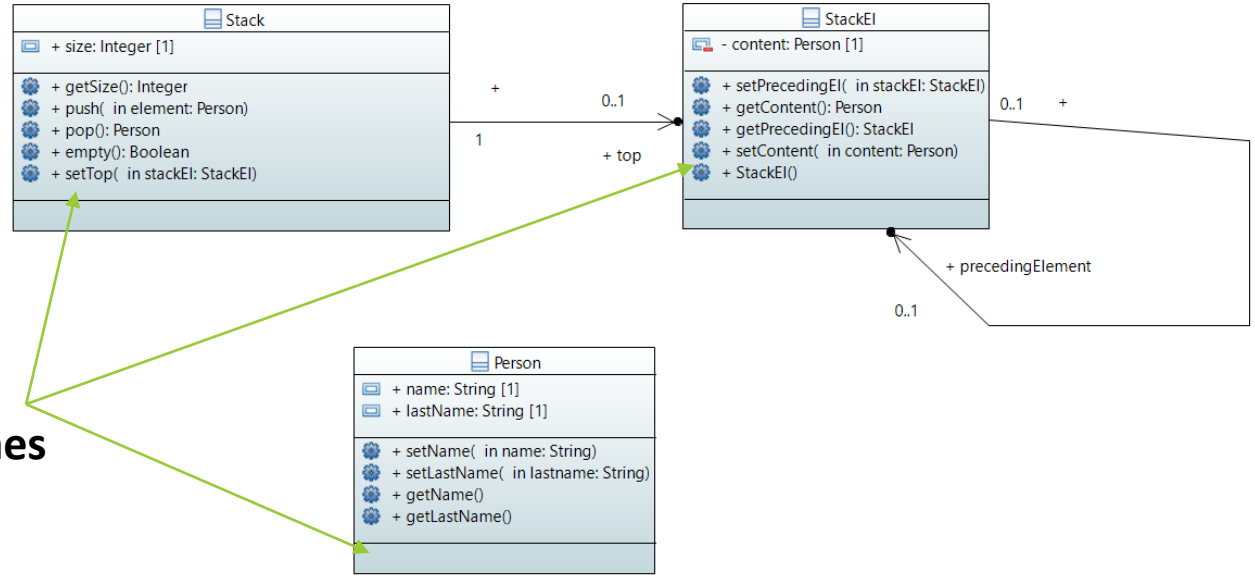


https://goo.gl/r37Pq4

Class diagram is *UML structure diagram* which shows structure of the designed system at the level of classes and interfaces, shows their properties, operations and relationships such associations, generalizations, dependencies, etc.

UML Class Diagrams

Definition



Operaciones

https://goo.gl/r37Pq4

Class diagram is *UML structure diagram* which shows structure of the designed system at the level of classes and interfaces, shows their properties, operations and relationships such associations, generalizations, dependencies, etc.

UML Class Diagrams

¿Vas a traducir de la misma forma la operación *setName* que *getName*?

UML Class Diagrams

Taxonomy of methods

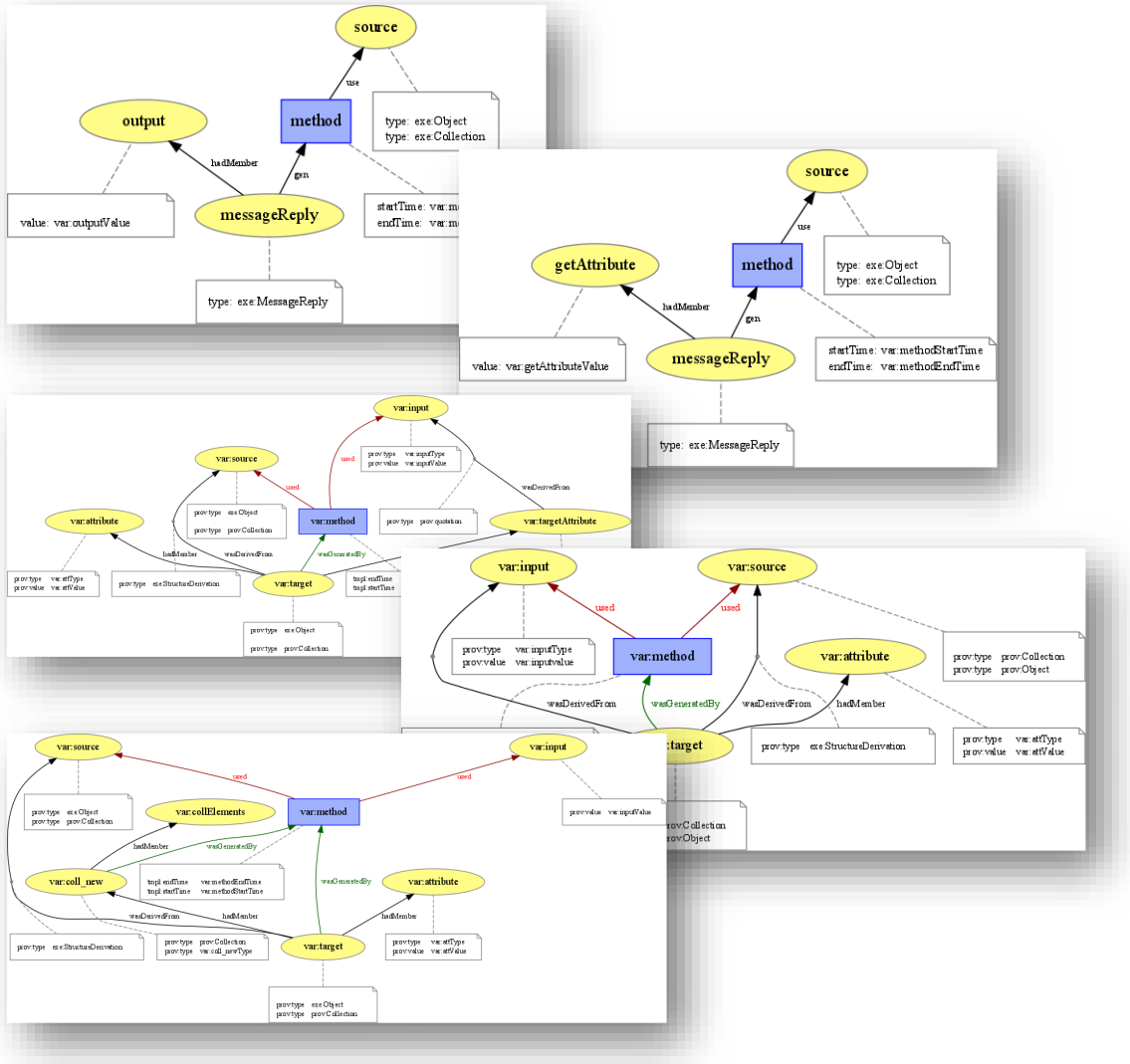
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member.
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member.
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory/Destroy	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…

Abid, N. J., Dragan, N., Collard, M. L., & Maletic, J. I. (2015, September). **Using stereotypes in the automatic generation of natural language summaries for C++ methods**. In *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on* (pp. 561-565). IEEE.

UML Class Diagrams

Taxonomy of method stereotypes

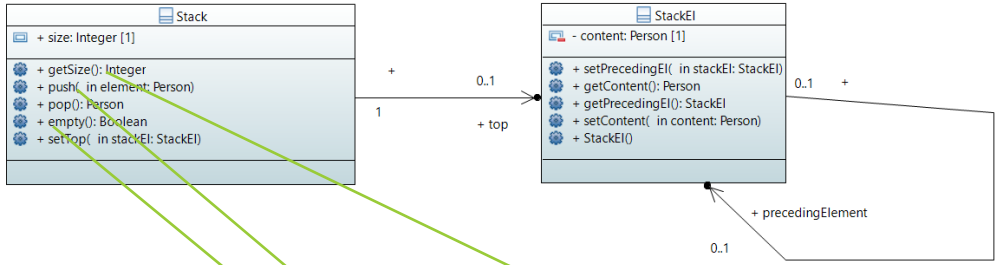
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member.
	Get-collection	**Returns an element from a data member collection.**
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member.
	Set-add-collection	**Adds an element within a data member collection.**
	Set-remove-collection	**Removes an element within a data member collection.**
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…

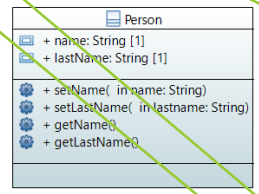


UML Class Diagrams

Linking methods with stereotypes

¿Cómo puedo saber a que categoría corresponde cada método?





Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member.
	Get-collection	**Returns an element from a data member collection.**
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member.
	Set-add-collection	**Adds an element within a data member collection.**
	Set-remove-collection	**Removes an element within a data member collection.**
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects....
	controller	Changes an external object's state...

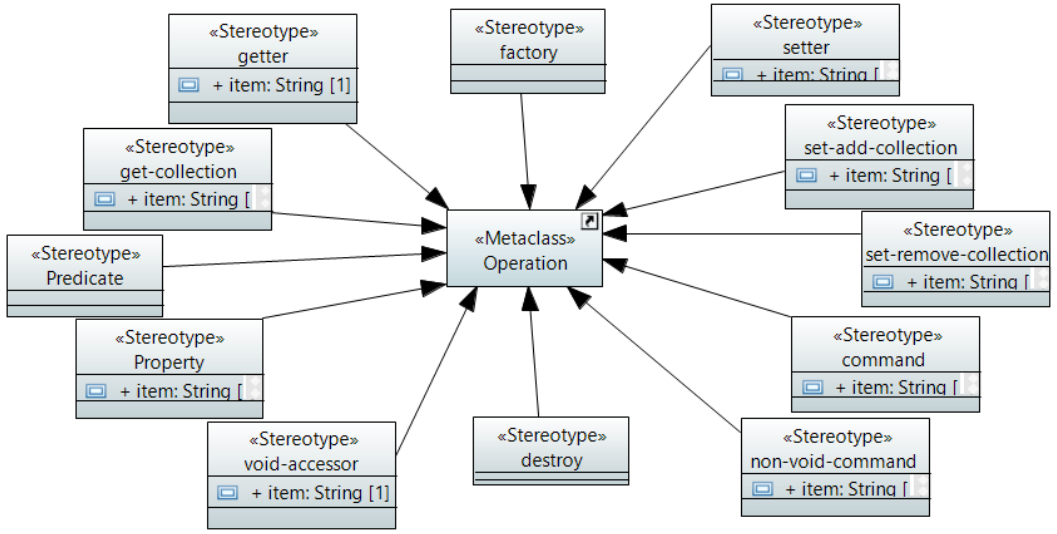
UML Class Diagrams

Linking methods with stereotypes – Option 1

¿Cómo puedo saber a que categoría corresponde cada método?

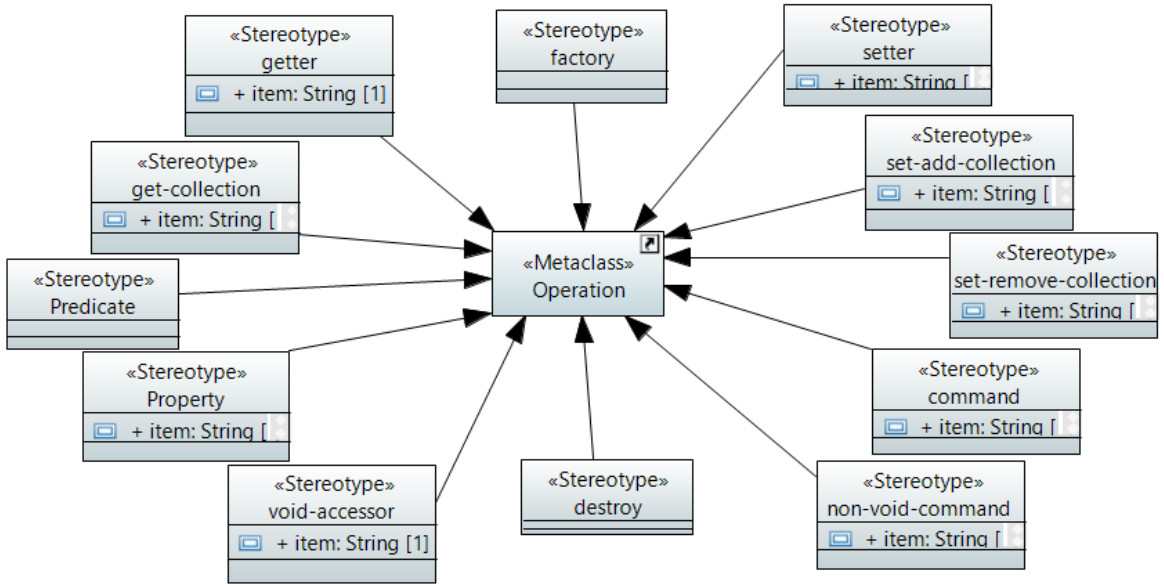
UML Stereotypes

A **stereotype** defines how an existing metaclass may be extended, and enables the use of platform or domain specific terminology or notation in place of, or in addition to, the ones used for the extended metaclass.



UML Class Diagrams

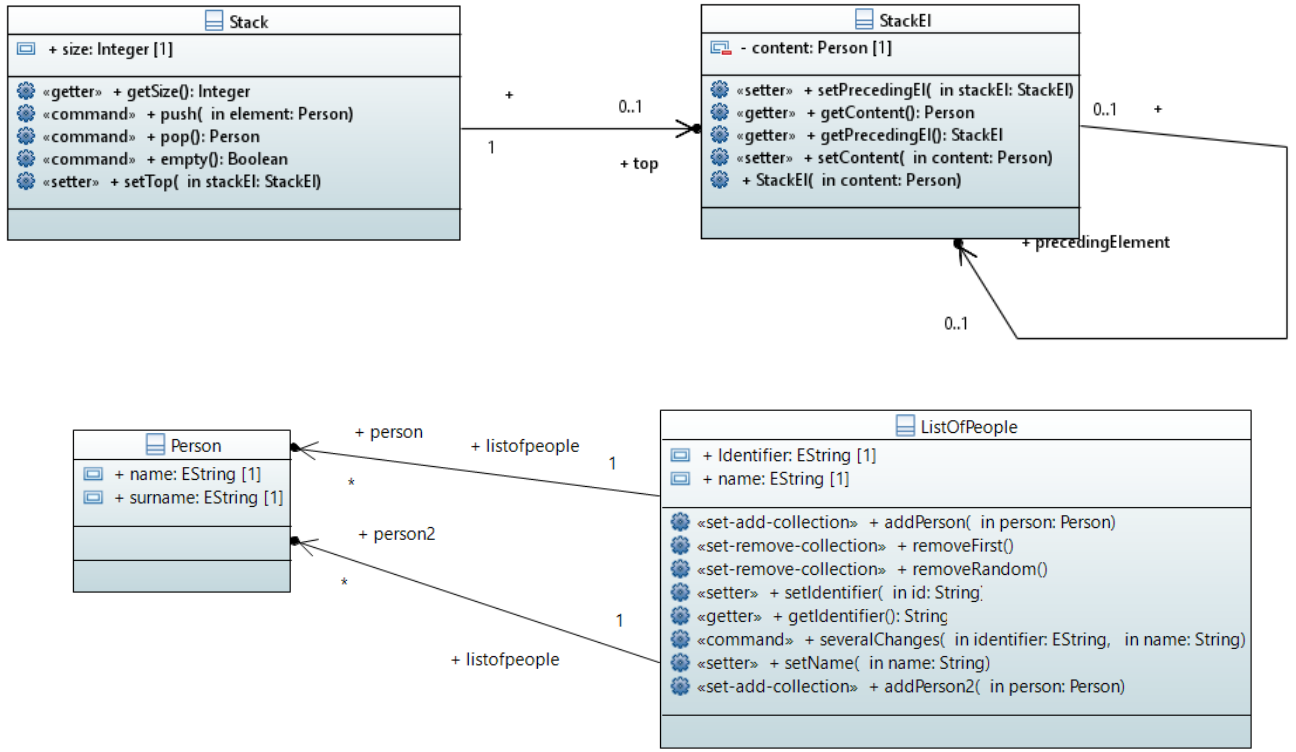
Linking methods with stereotypes – Option 1



Stereotype Category	Method Stereotype	UML Stereotype
Structural Accessor	Get	getter
	Get-collection	**get-collection**
	Predicate	prediate
	Property	property
	Void-accessor	void-accessor
Structural Mutator	Set	setter
	Set-add-collection	**set-add-collection**
	Set-remove-collection	**set-remove-collection**
	Command	command
	Non-void-command	Non-void-command
Creational	Factory/Destroy	factory / destroy
Collaborational	collaborator	
	controller	

UML Class Diagrams

Linking methods with stereotypes – Option 1



UML Class Diagrams

Linking methods with stereotypes – Option 2

Stereotype Category	Method Stereotype	Operation Signature
Structural Accessor	Get	get<Property>
	Get-collection	
	Predicate	
	Property	
	Void-accessor	
Structural Mutator	Set	set<Property>
	Set-add-collection	**add<Property>**
	Set-remove-collection	**remove<Property>**
	Command	
	Non-void-command	
Creational	Factory/Destroy	<className>
Collaborational	collaborator	
	controller	

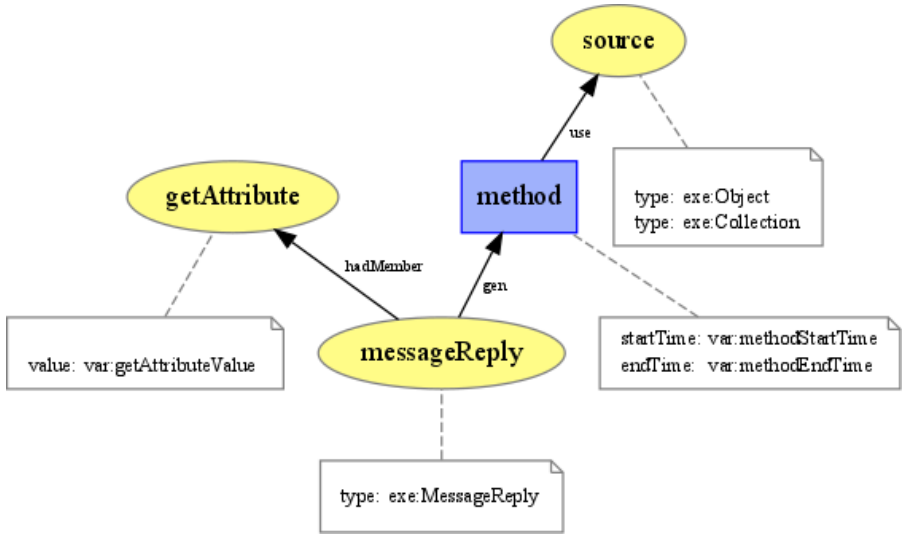
- getSize(…)
- getName (…)
- getContent (…)

- setName (…)
- setIdentifier (…)
- addPerson (…)
- removePerson (…)

UML Class Diagrams

Translation of method stereotypes – Structural Accessor

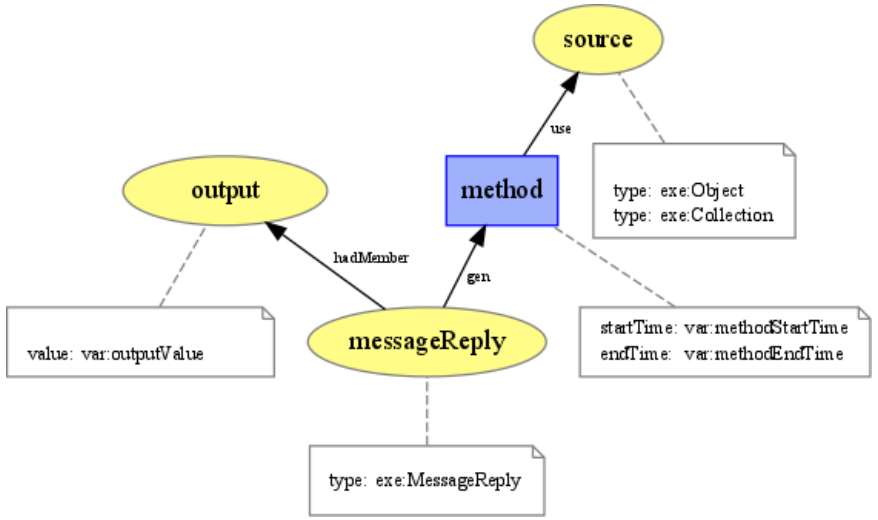
Stereotype Category	Stereotype	Description
Structural Accessor	**Get**	**Returns a data member**
	Get-collection	**Returns an element from a data member collection**
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member
	Set-add-collection	Adds an element within a data member collection.
	Set-remove-collection	Removes an element within a data member collection.
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



UML Class Diagrams

Translation of method stereotypes - Structural Accessor

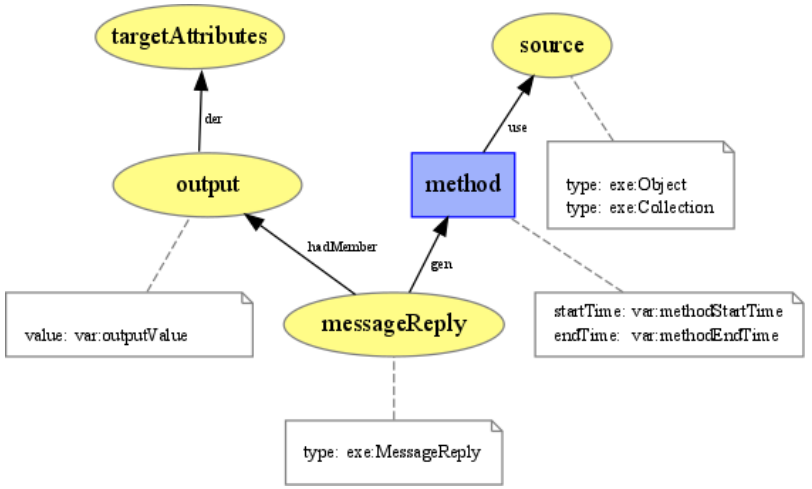
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member
	Get-collection	Returns an element from a data member collection
	Predicate	**Returns a Boolean value which is not a data member.**
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member
	Set-add-collection	Adds an element within a data member collection.
	Set-remove-collection	Removes an element within a data member collection.
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



UML Class Diagrams

Translation of method stereotypes - Structural Accessor

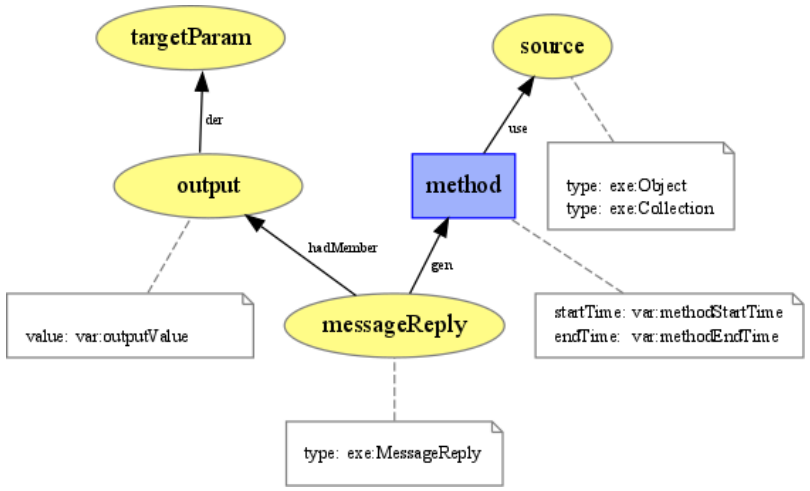
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member
	Get-collection	Returns an element from a data member collection
	Predicate	Returns a Boolean value which is not a data member.
	Property	**Returns information about data members.**
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member
	Set-add-collection	Adds an element within a data member collection.
	Set-remove-collection	Removes an element within a data member collection.
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



UML Class Diagrams

Translation of method stereotypes – Structural Accessor

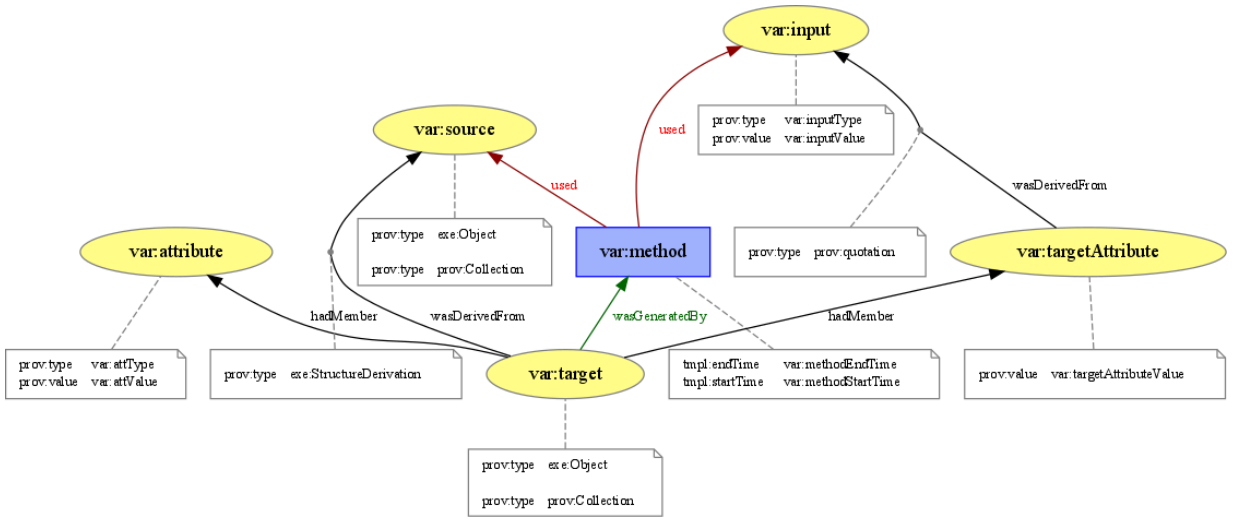
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member
	Get-collection	Returns an element from a data member collection
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	**Returns information through a parameter.**
Structural Mutator	Set	Sets a data member
	Set-add-collection	Adds an element within a data member collection.
	Set-remove-collection	Removes an element within a data member collection.
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



UML Class Diagrams

Translation of method stereotypes – Void-accessor

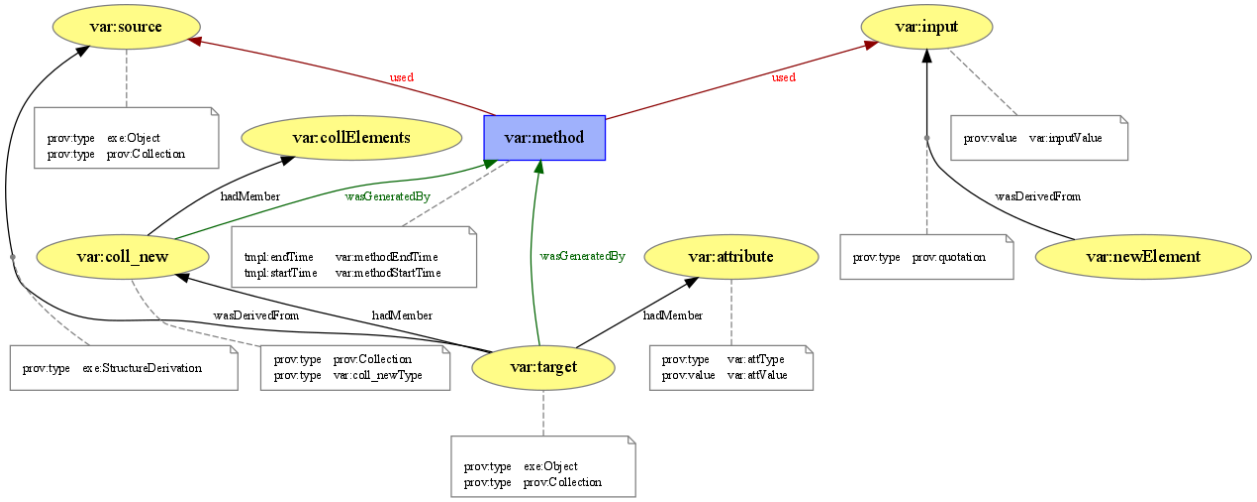
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member
	Get-collection	Returns an element from a data member collection
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	**Set**	**Sets a data member**
	Set-add-collection	Adds an element within a data member collection.
	Set-remove-collection	Removes an element within a data member collection.
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



UML Class Diagrams

Translation of method stereotypes – Set-add-collection

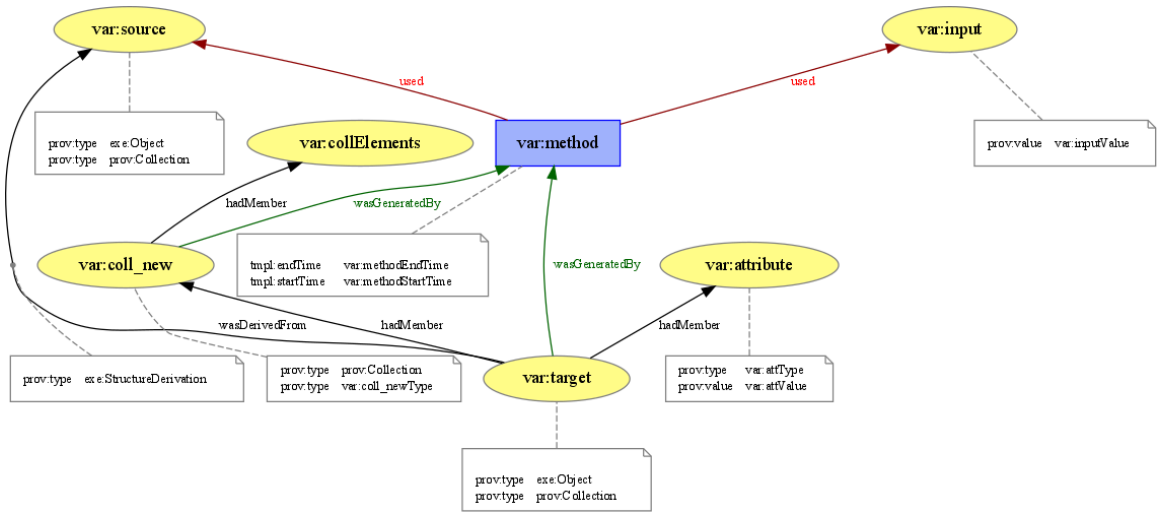
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member
	Get-collection	Returns an element from a data member collection
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member
	Set-add-collection	**Adds an element within a data member collection.**
	Set-remove-collection	Removes an element within a data member collection.
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



UML Class Diagrams

Translation of method stereotypes – Set-remove-collection

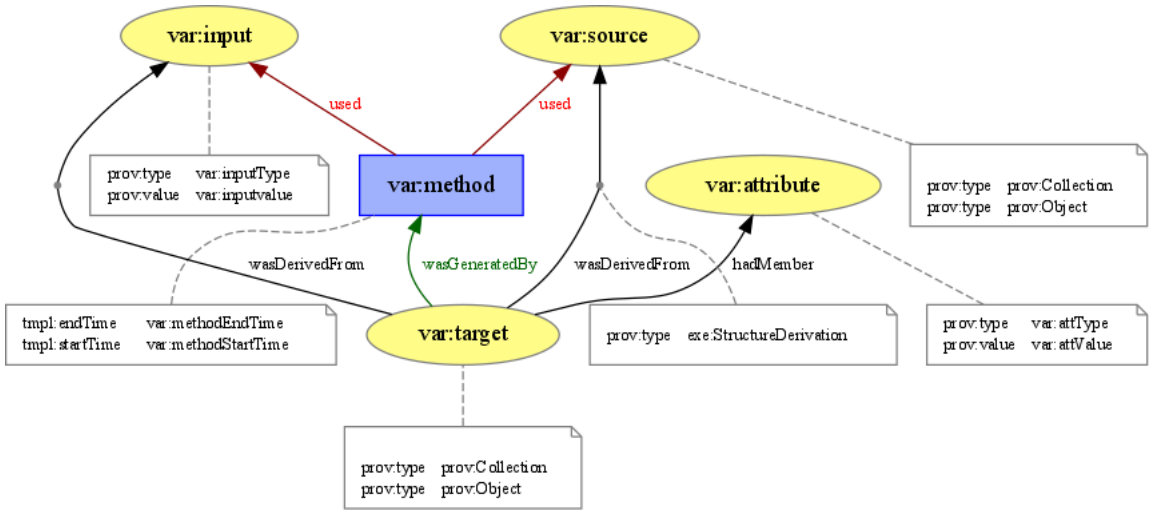
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member
	Get-collection	Returns an element from a data member collection
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member
	Set-add-collection	Adds an element within a data member collection.
	Set-remove-collection	**Removes an element within a data member collection.**
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



UML Class Diagrams

Translation of method stereotypes – Command

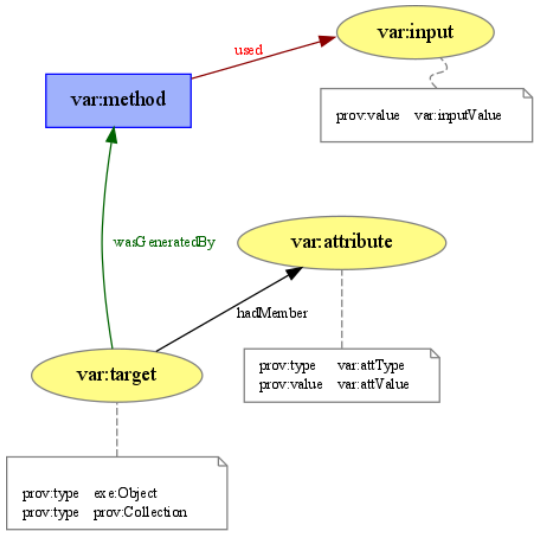
Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member
	Get-collection	Returns an element from a data member collection
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member
	Set-add-collection	Adds an element within a data member collection.
	Set-remove-collection	Removes an element within a data member collection.
	Command	**Performs complex change to an object.**
	Non-void-command	**Performs complex change to an object.**
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



UML Class Diagrams

Translation of method stereotypes – Factory

Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member
	Get-collection	Returns an element from a data member collection
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member
	Set-add-collection	Adds an element within a data member collection.
	Set-remove-collection	Removes an element within a data member collection.
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	**Factory**	**Creates and/or destroys objects.**
Collaborational	collaborator	Works with objects….
	controller	Changes an external object's state…



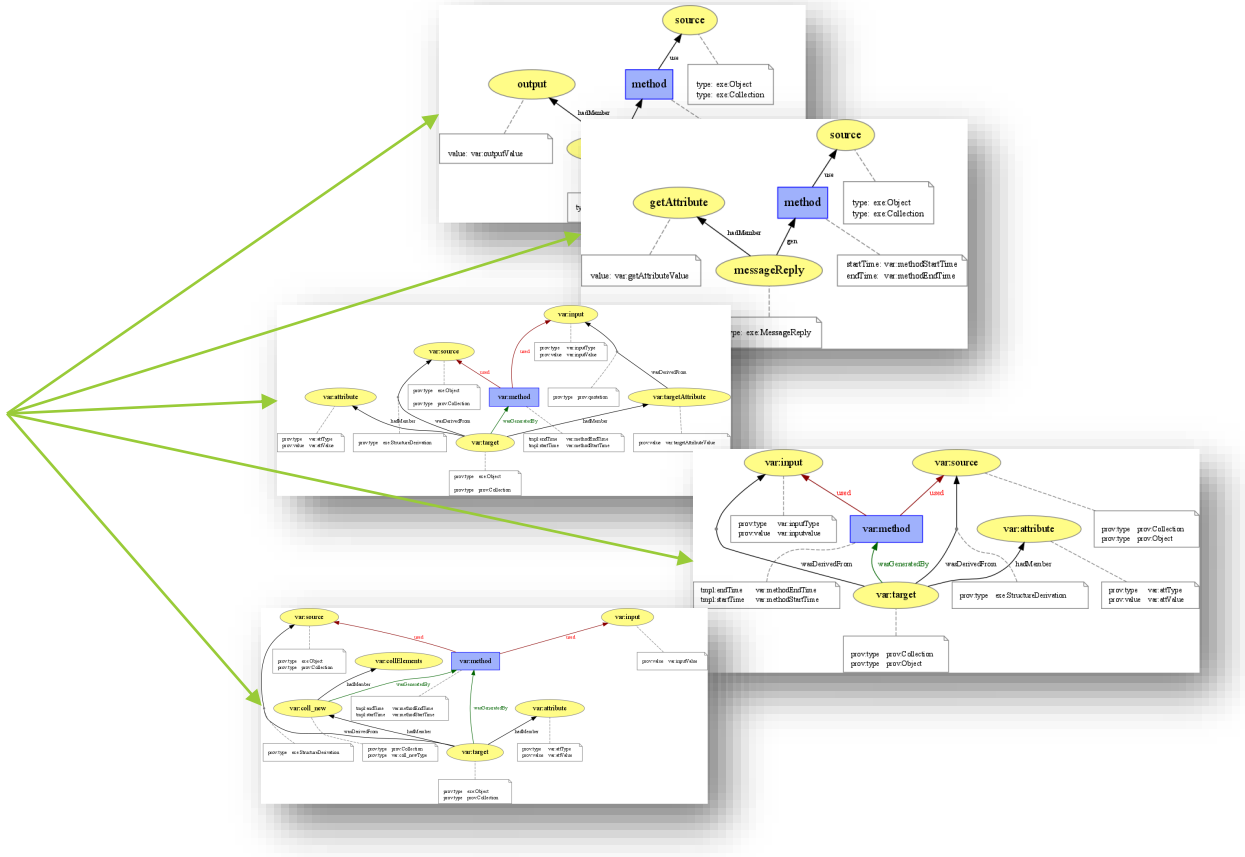
UML Class Diagrams

Taxonomy of method stereotypes

Taxonomy of operation

Stereotype Category	Stereotype	Description
Structural Accessor	Get	Returns a data member.
	Get-collection	**Returns an element from a data member collection.**
	Predicate	Returns a Boolean value which is not a data member.
	Property	Returns information about data members.
	Void-accessor	Returns information through a parameter.
Structural Mutator	Set	Sets a data member.
	Set-add-collection	**Adds an element within a data member collection.**
	Set-remove-collection	**Removes an element within a data member collection.**
	Command	Performs complex change to an object.
	Non-void-command	Performs complex change to an object.
Creational	Factory	Creates and/or destroys objects.
Collaborational	collaborator	Works with objects....
	controller	Changes an external object's state...

PROV templates



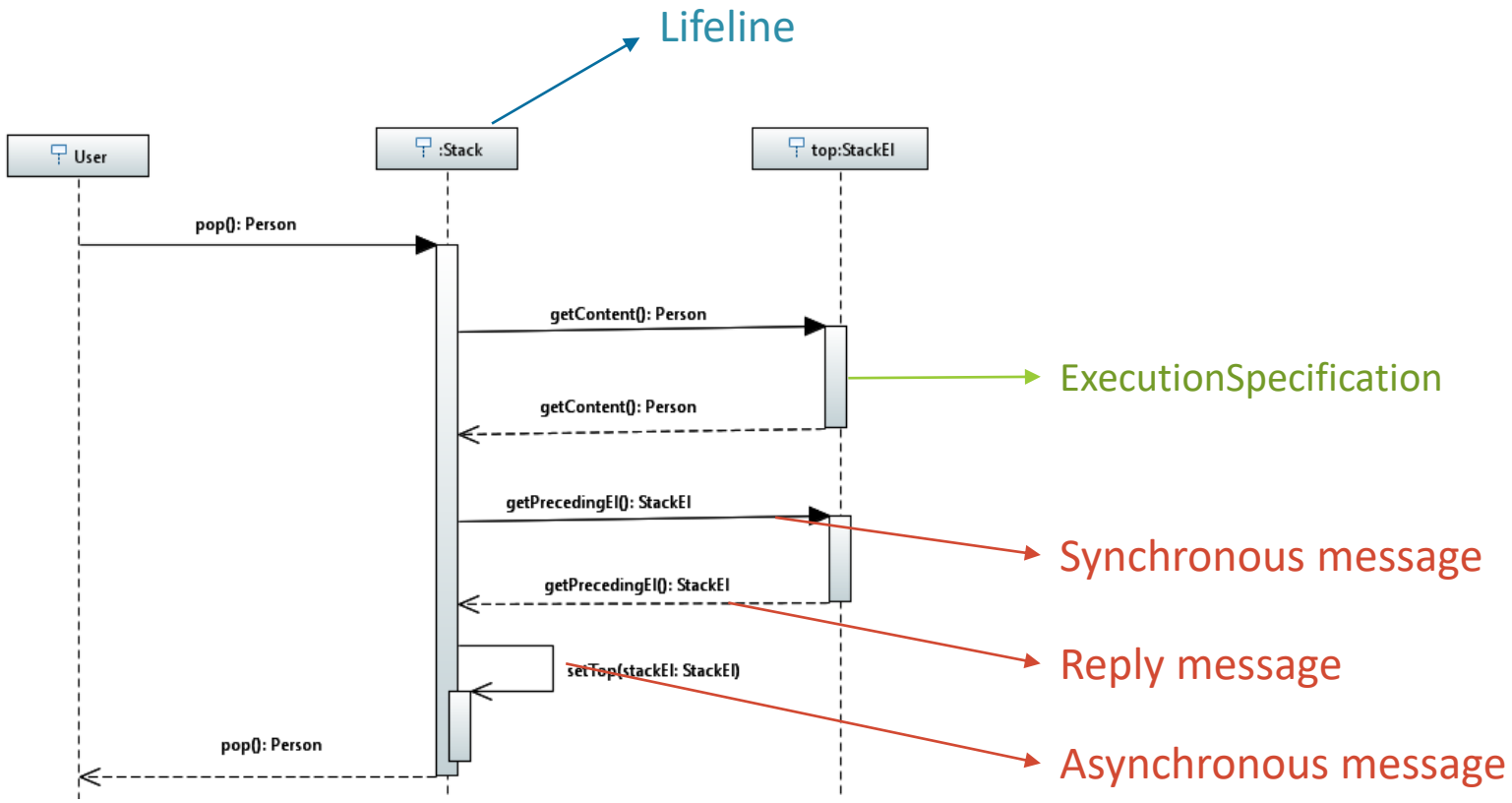
Índice

- Definición de *provenance*
- Ejemplos de *provenance*
- W3C PROV standard
- PROV-Templates
- Methodology overview
- **De UML a PROV**
  - UML Class Diagrams
  - **UML Sequence Diagrams (revisited)**
- Aplicación de la programación orientada a aspectos para la generación de *bindings.*
- Ejemplo

UML Sequence Diagrams

Definition

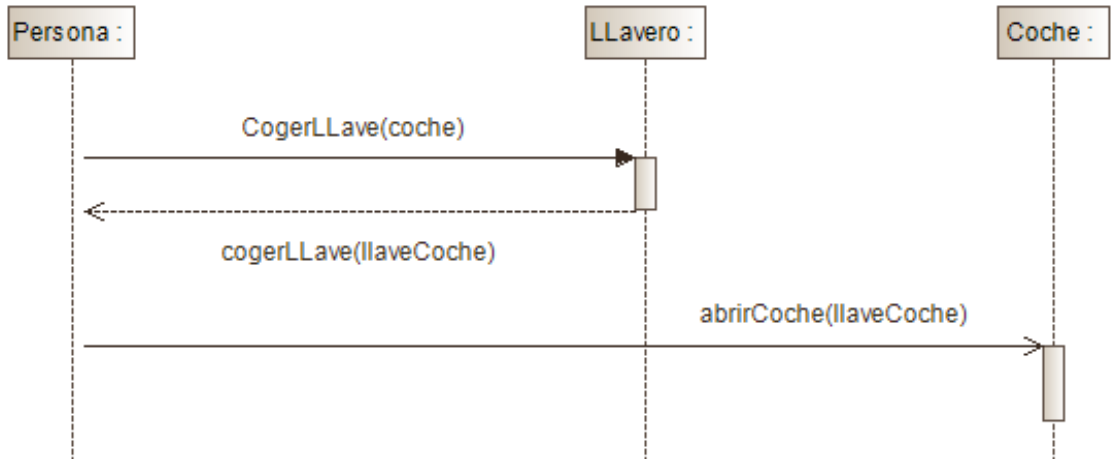
Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines.

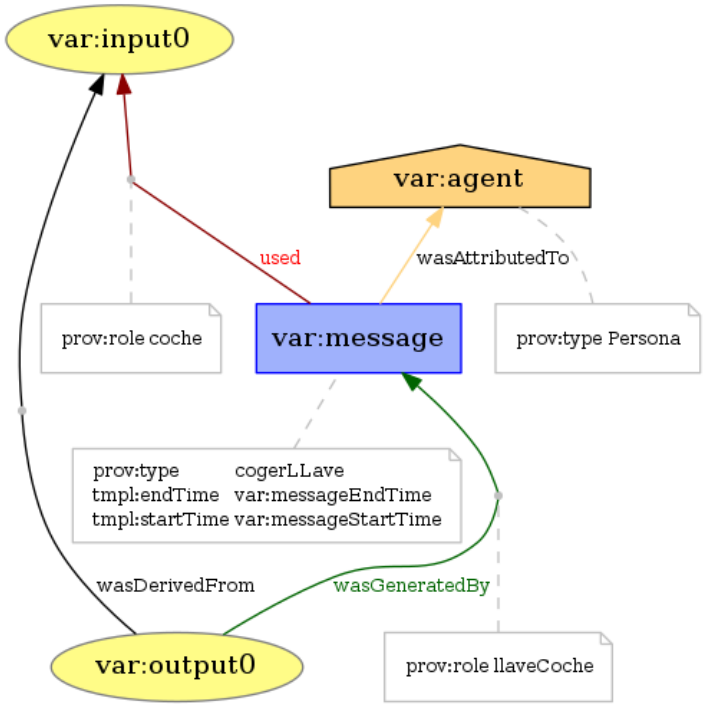


https://www.uml-diagrams.org/sequence-diagrams.html

UML Sequence Diagrams

v1

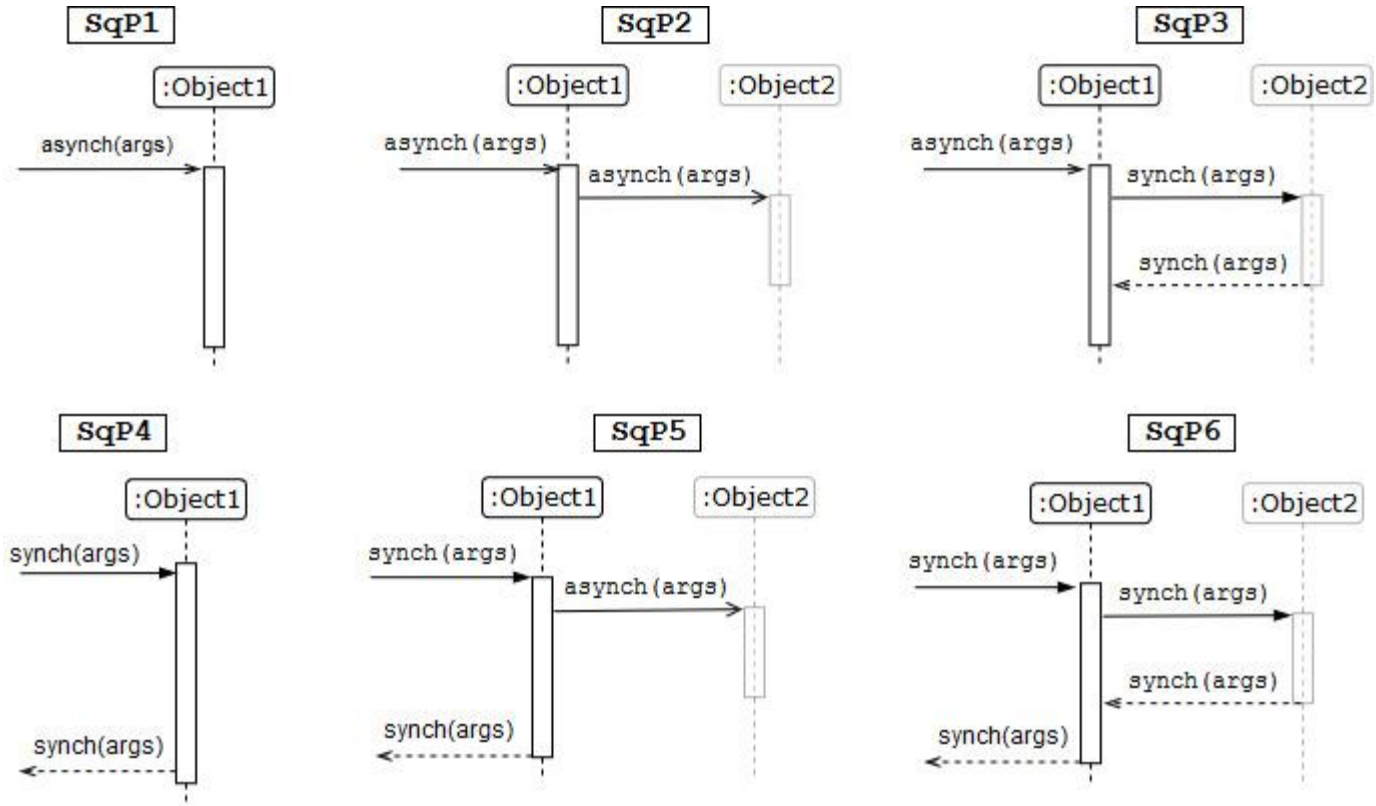




https://www.uml-diagrams.org/sequence-diagrams.html

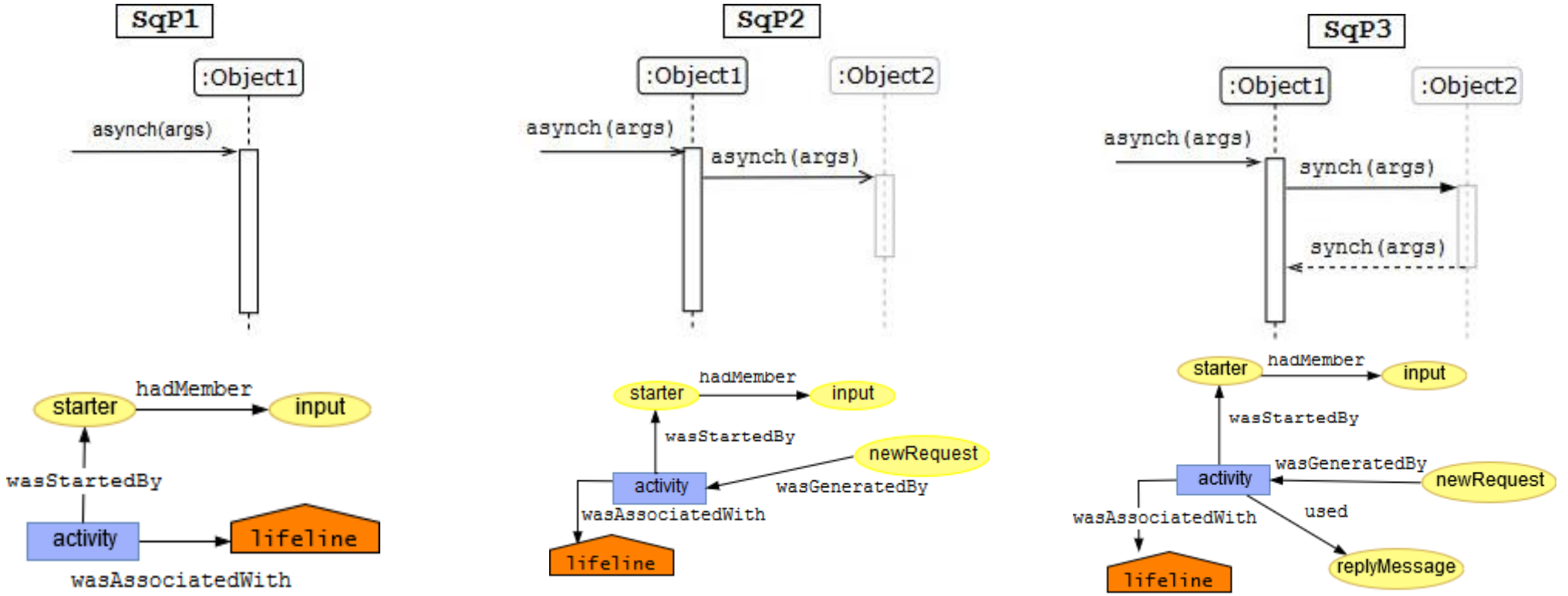
UML Sequence Diagrams

Set of Patterns Identified



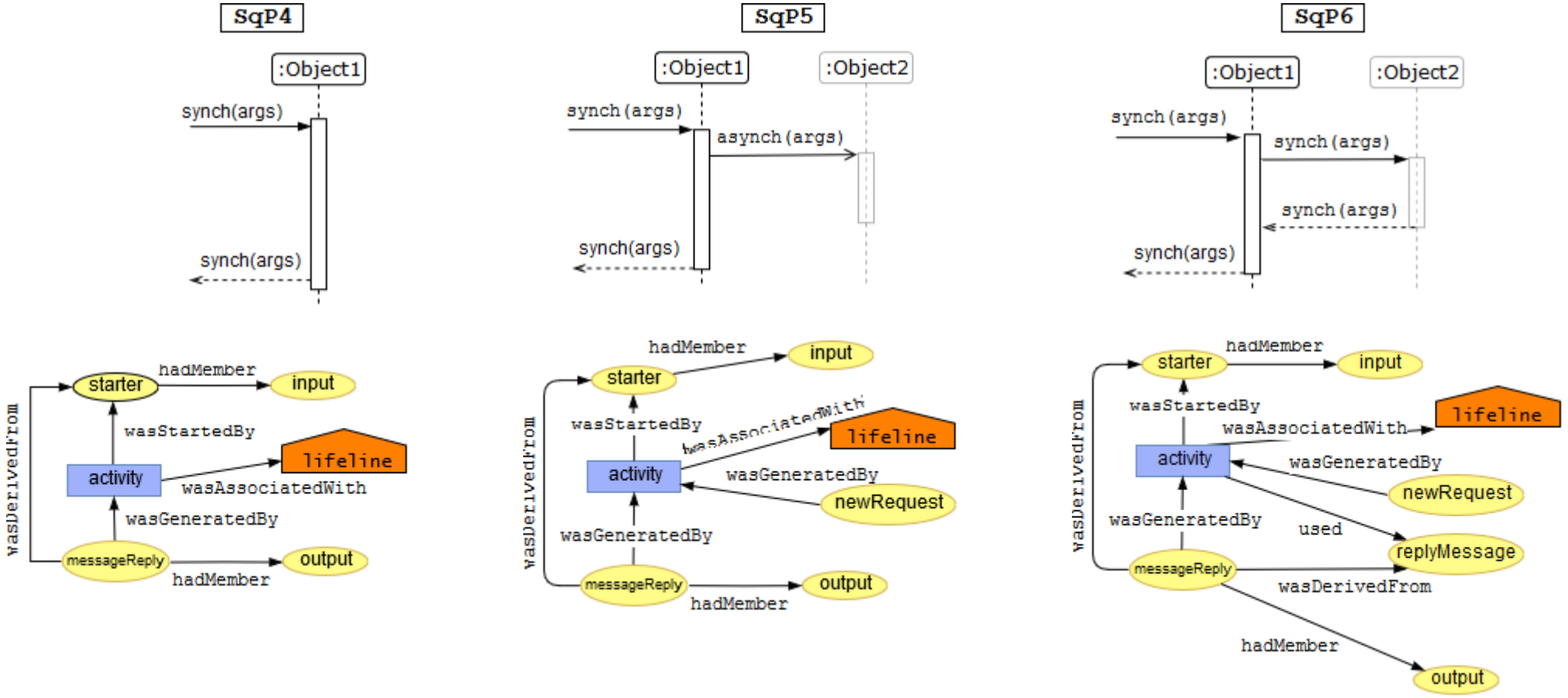
UML Sequence Diagrams

Set of Transformation



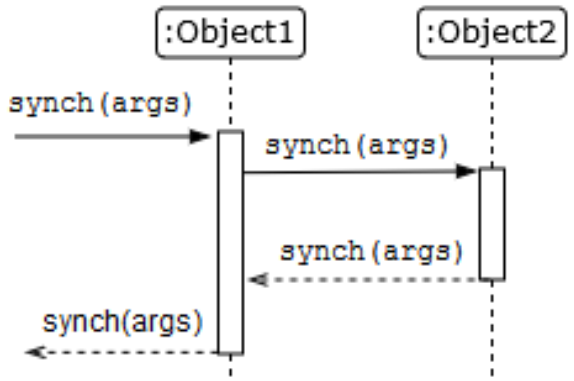
UML Sequence Diagrams

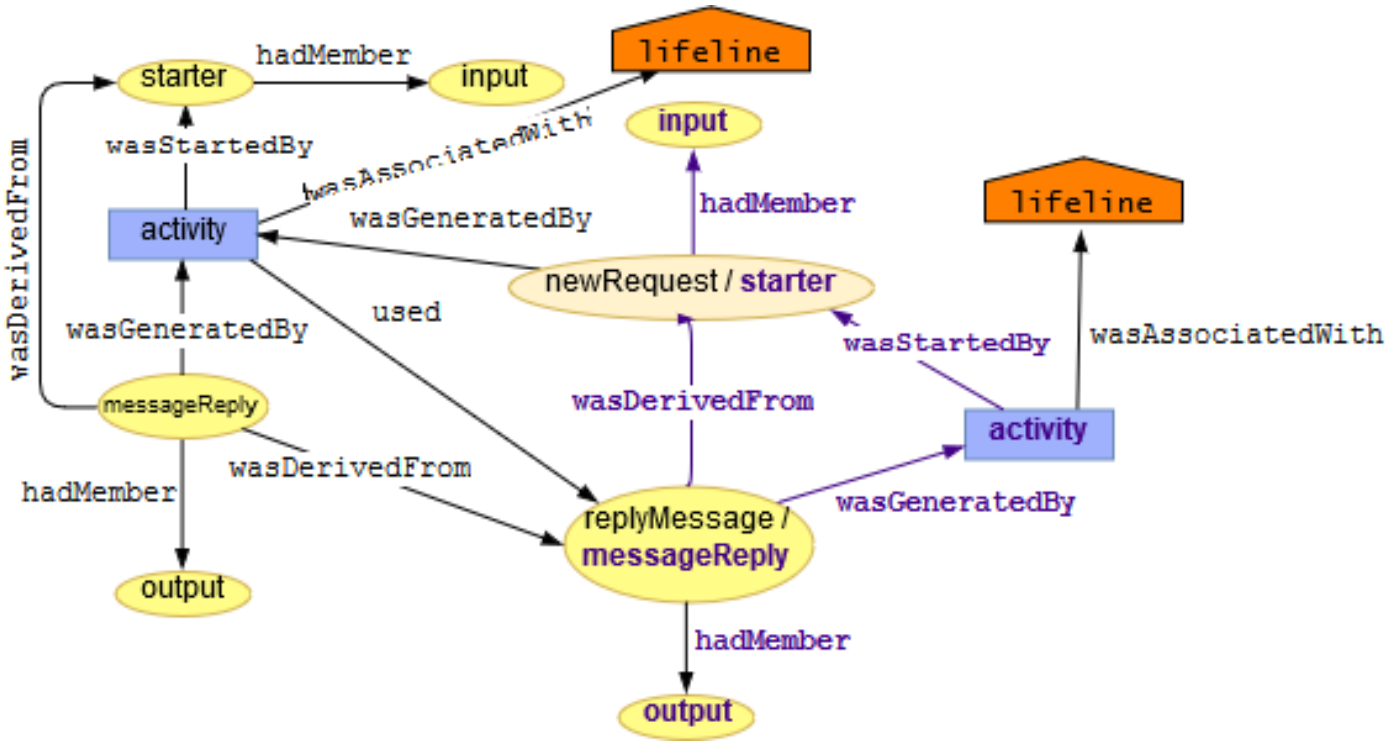
Set of Patterns Identified



UML Sequence Diagrams

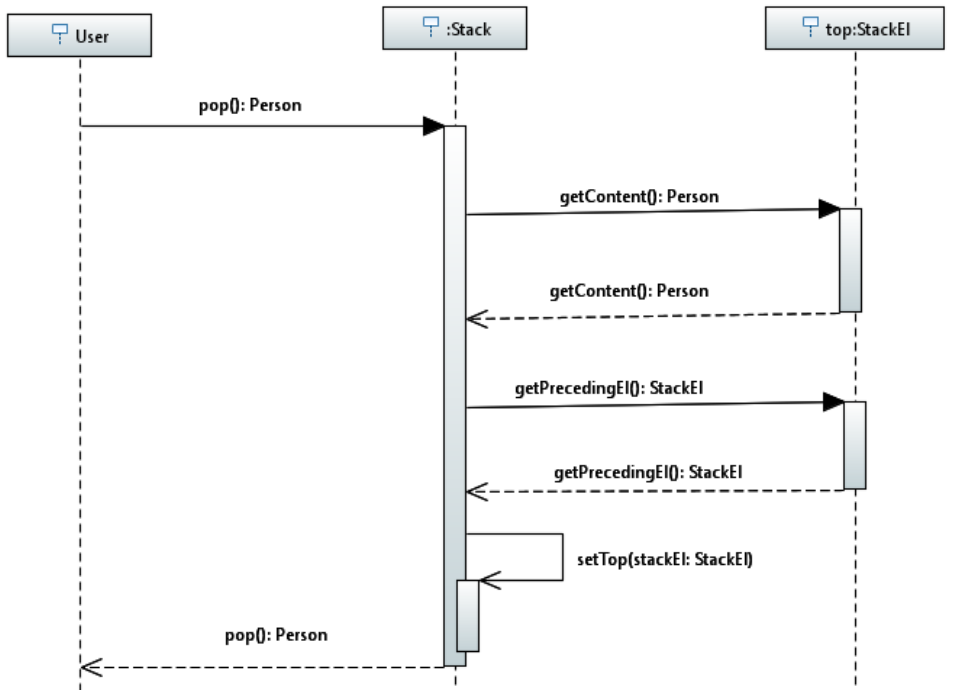
Interconnection between activities

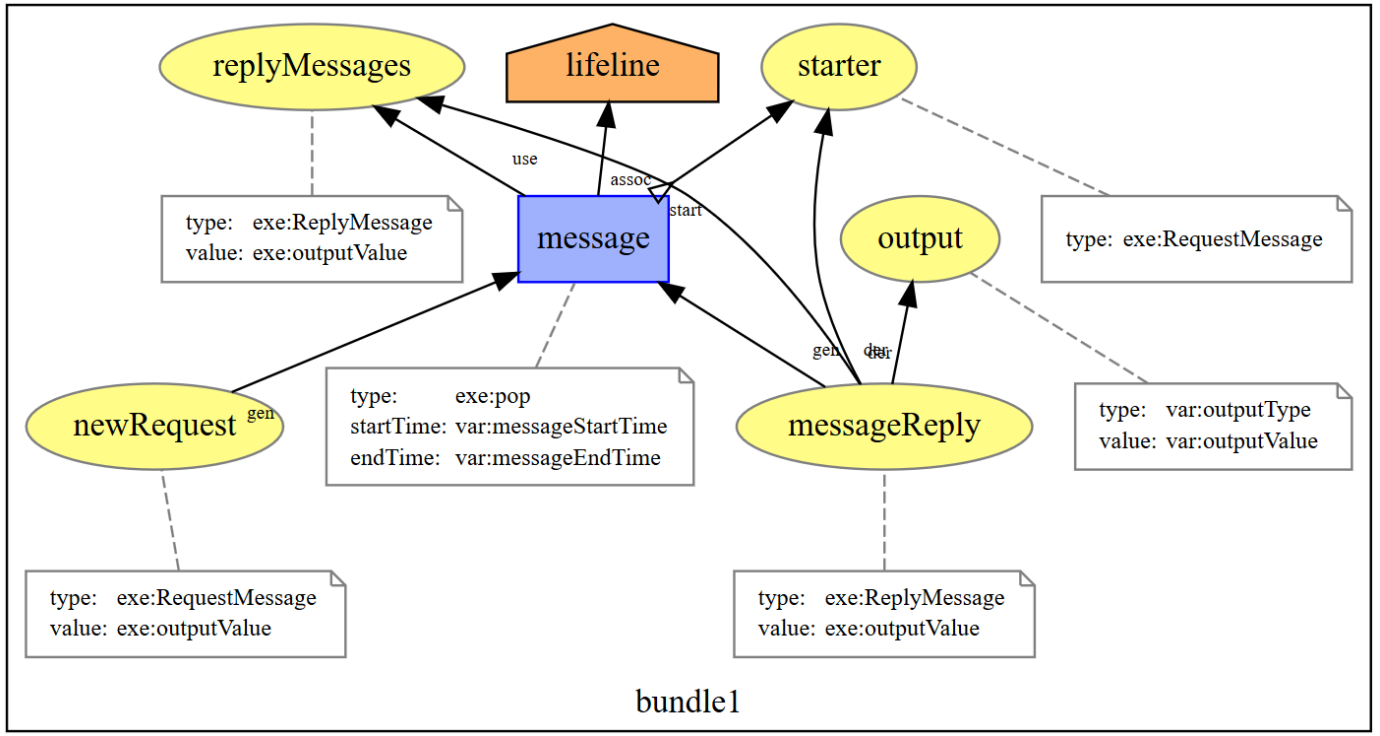




UML Sequence Diagrams

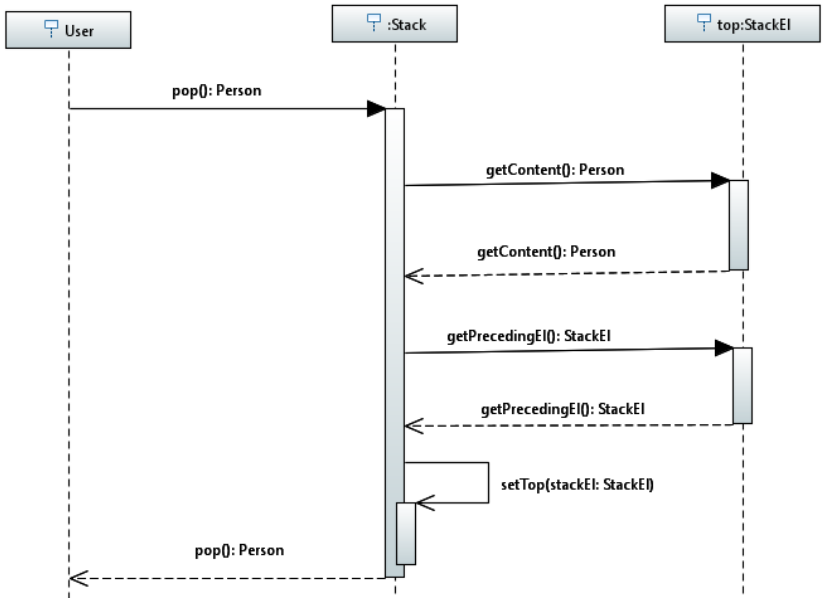
Example





Índice

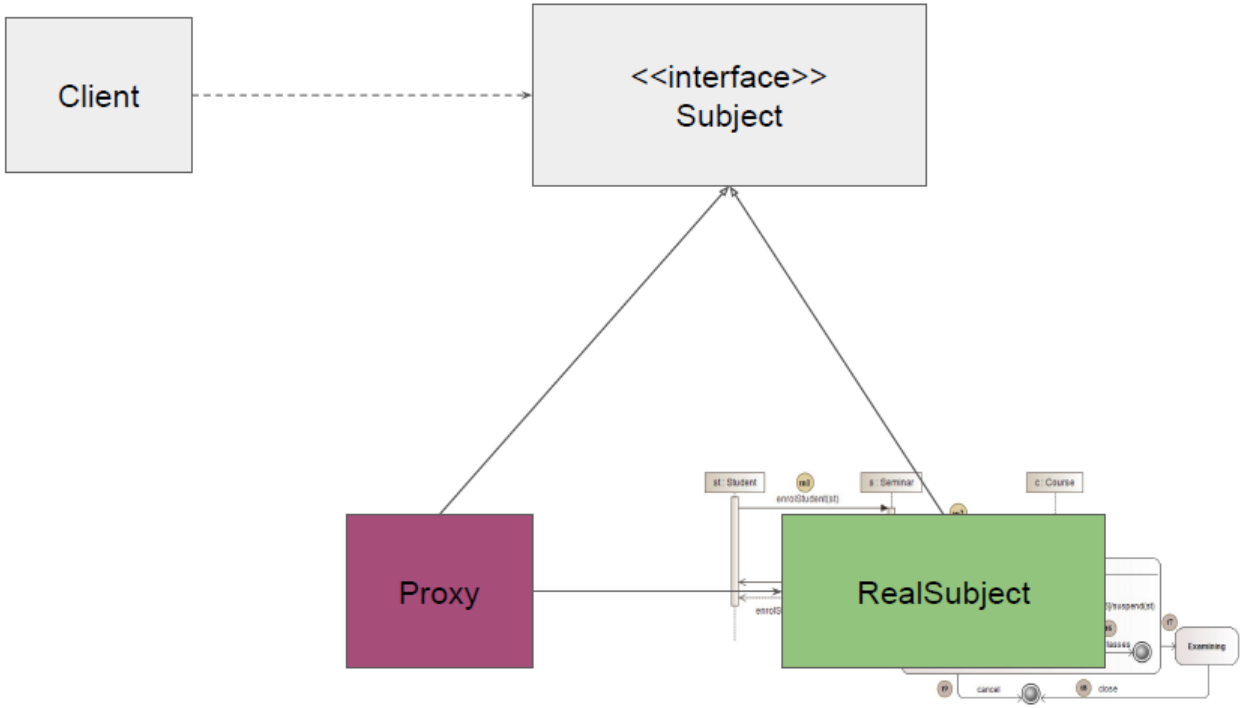
Generación de *bindings*




```
document
prefix tmpl <http://openprovenance.org/tmpl#>
prefix var <http://openprovenance.org/var#>
prefix exe <http://example.org/>

    entity(var:starter,[tmpl:value_0 = 'exe:476ff0b9-40d8-448c-9706-4f39155a1241'])
    entity(var:source,[tmpl:value_0 = 'exe:608a69da-401d-4702-a1c3-f5e7c23384a4_2'])

    entity(var:messageReply,[tmpl:value_0 = 'exe:8a018b3f-0930-47d7-b18d-bcf1d4f52041'])
    entity(var:outputType,[tmpl:2dvalue_0_0 = 'exe:RootElement.Person@5ebec15'])
    entity(var:outputValue,[tmpl:2dvalue_0_0 = "RootElement.Person@5ebec15" %% xsd:string])
    entity(var:output,[tmpl:value_0 = 'exe:e9c08b73-d254-4cab-9298-6efcdc36ada2_1'])

    entity(var:method,[tmpl:value_0 = 'exe:pop_608a69da-401d-4702-a1c3-f5e7c23384a4_2_e483e15e-7279-4735-8c86-e165032cc0a2'])
    entity(var:methodStartTime,[tmpl:2dvalue_0_0 = "2017-12-18T13:04:46" %% xsd:dateTime])
    entity(var:methodEndTime,[tmpl:2dvalue_0_0 = "2017-12-18T13:04:46" %% xsd:dateTime])
    entity(var:target,[tmpl:value_0 = 'exe:608a69da-401d-4702-a1c3-f5e7c23384a4_3'])

    entity(var:object,[tmpl:value_0 = 'exe:608a69da-401d-4702-a1c3-f5e7c23384a4'])
    entity(var:lifeline,[tmpl:value_0 = 'exe:Principal'])
    entity(var:objectSMD,[tmpl:value_0 = 'exe:608a69da-401d-4702-a1c3-f5e7c23384a4_0'])

    entity(var:attribute,[tmpl:value_0 = 'exe:79a35047-b97e-49da-b29c-0ba280ea8a9b',
        tmpl:value_1 = 'exe:7e66088e-1037-447e-9235-e5d5c45fe52a'])
    entity(var:attType,[tmpl:2dvalue_0_0 = "class StackExample.StackEl" %% xsd:string,
        tmpl:2dvalue_1_0 = "int" %% xsd:string])
    entity(var:attValue,[tmpl:2dvalue_0_0 = "null" %% xsd:string,
        tmpl:2dvalue_1_0 = "0" %% xsd:string])

    entity(var:replyMessages,[tmpl:value_0 = 'exe:2029cad8-5f0c-43e5-9c98-83d947597d15',
        tmpl:value_1 = 'exe:c4351b25-fe88-419e-8548-8e70f60a45d4',
        tmpl:value_2 = 'exe:88659c77-e043-431d-95bb-25395b7f90fb'])
    entity(var:newRequest,[tmpl:value_0 = 'exe:b8704ab6-99c3-4db7-8e1c-3ef7ed1b80f9',
        tmpl:value_1 = 'exe:f4ddc703-1d2a-404a-be28-4e14ecd06916',
        tmpl:value_2 = 'exe:205ab5ee-9c65-4329-968d-d200606b9ad6'])

endDocument
```

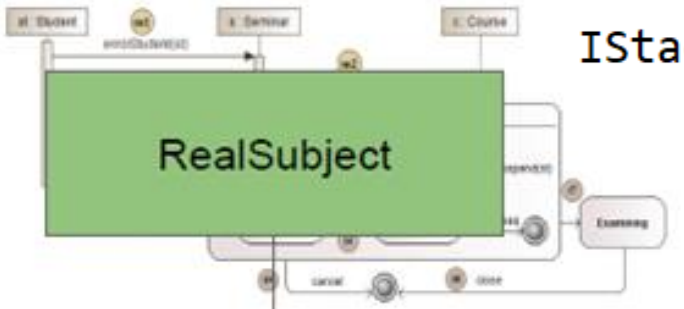
Generación de *bindings*

Proxy-Pattern



Generación de *bindings*

Proxy-Pattern




```
IStack st = new Stack();
```

ProxyProv

```
st = ProxyProvGenerator.generateProxy(st);
```

Proxy

Bindings

Generación de *bindings*

Proxy-Pattern

Puntos débiles

- No permite capturar la invocación de un constructor.
- Hay que modificar el código fuente.

Generación de *bindings*

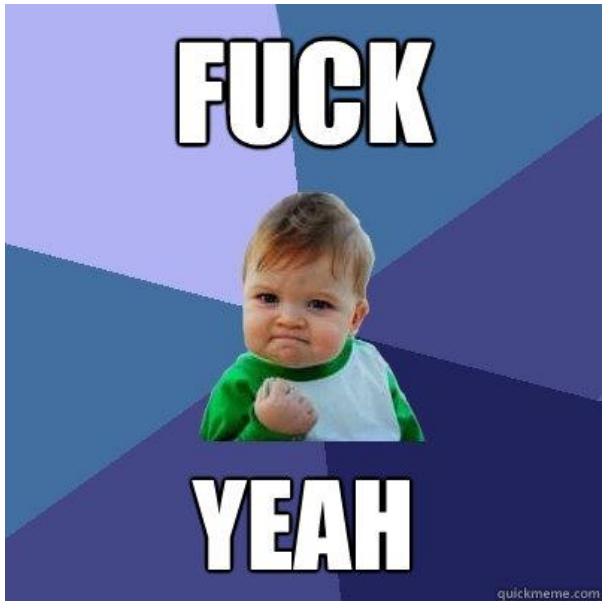
Programación Orientada a Aspectos

Aspect-Oriented Programming (AOP) es un paradigm de programación que busca proporcionar al programador una forma “limpia” de separar intereses (o *concerns*) transversales. Esto se consigue añadiendo un comportamiento adicional en el código ya existente sin realizar ningún tipo de modificaciones.

Generación de *bindings*

Programación Orientada a Aspectos

Aspect-Oriented Programming (AOP) es un paradigm de programación que busca proporcionar al programador una forma “limpia” de separar intereses (o *concerns*) transversales. Esto se consigue añadiendo un comportamiento adicional en el código ya existente SIN reslizar ningún tipo de MODIFICACIONES.



Generación de *bindings*

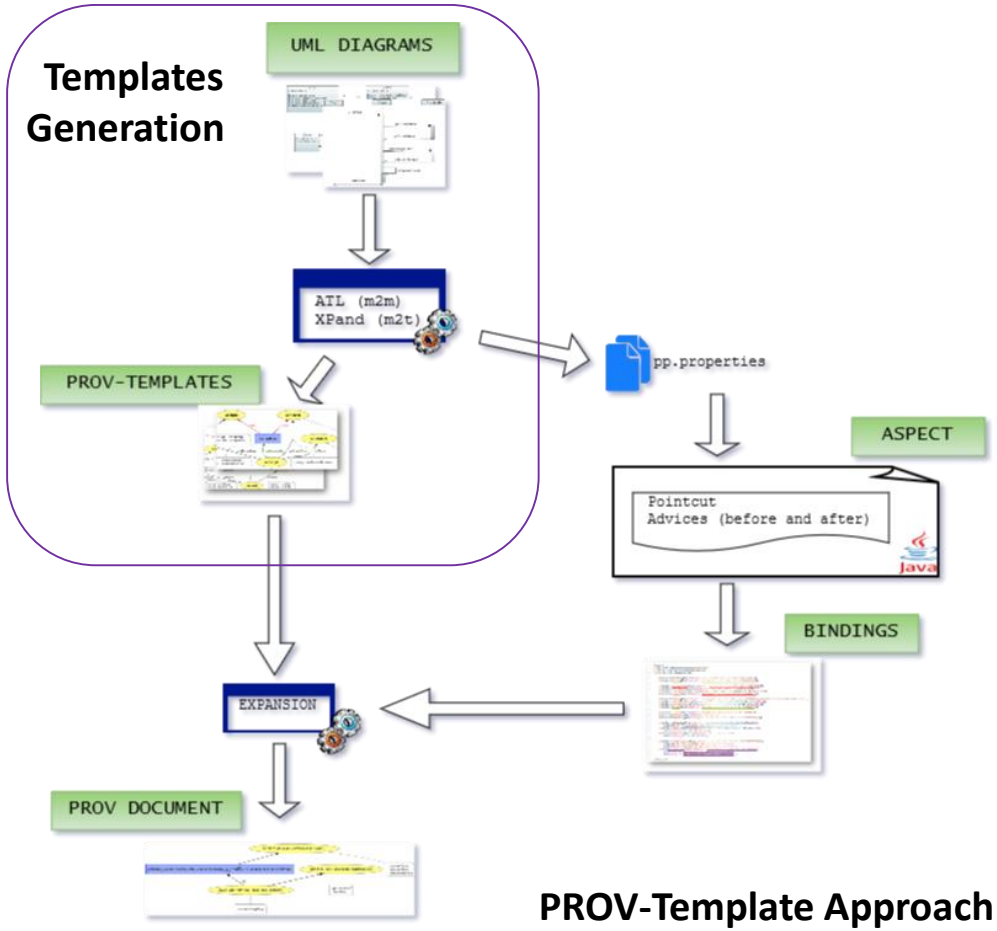
Programación Orientada a Aspectos

1. Capturo la llamada a un método o a un constructor a través de un *pointcut*.
2. Ejecuto código para la generación de bindings (*before advice*).
3. Ejecuto el comportamiento normal del método.
4. Ejecuto código para la generación de bindings (*after advice*).
5. Devuelvo el flujo del programa a su comportamiento normal.

Índice

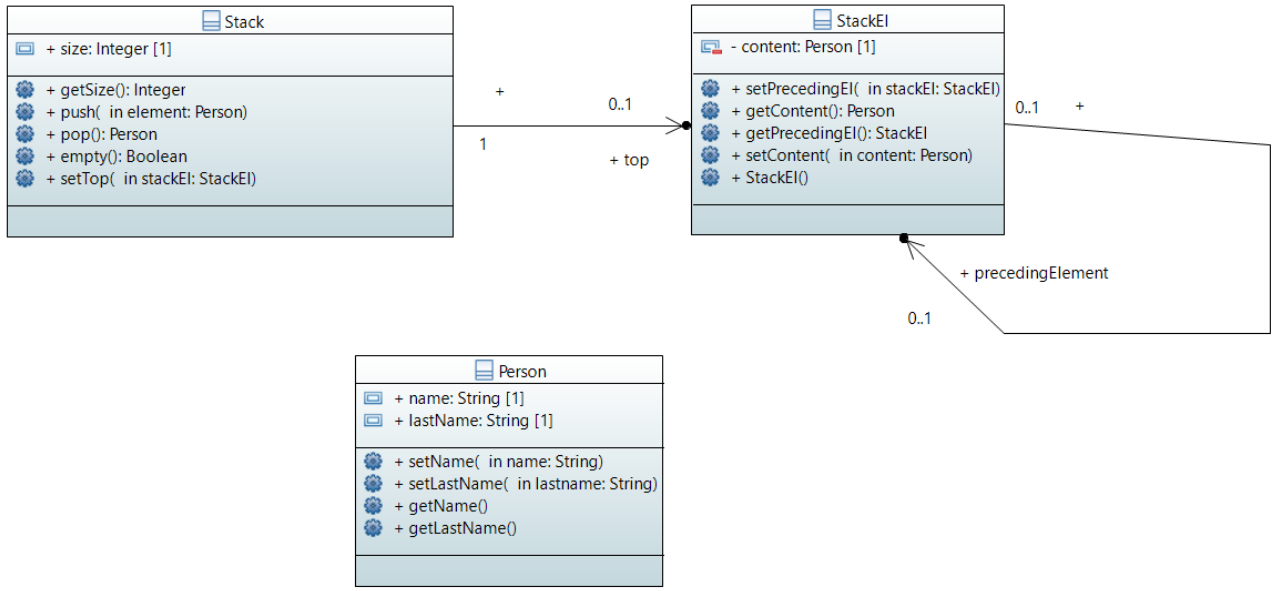
Ejemplo

UML diagrams used



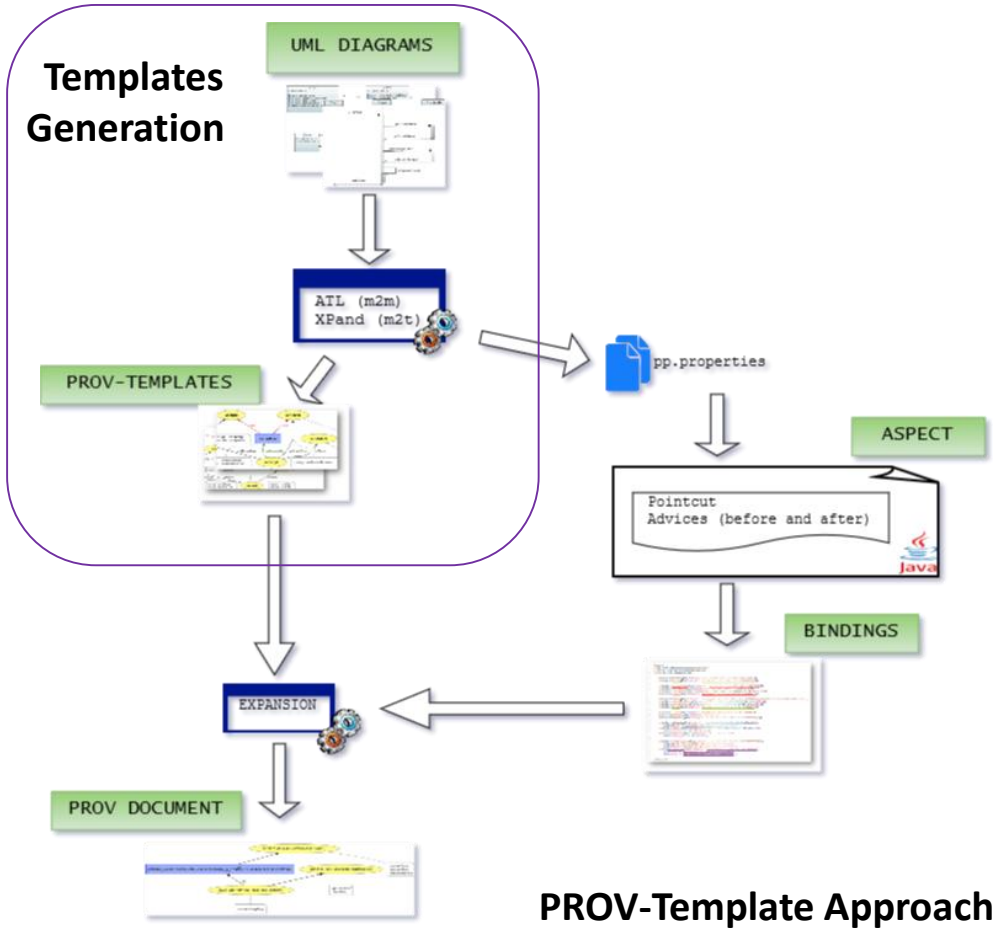
PROV-Template Approach

UML Diagrams

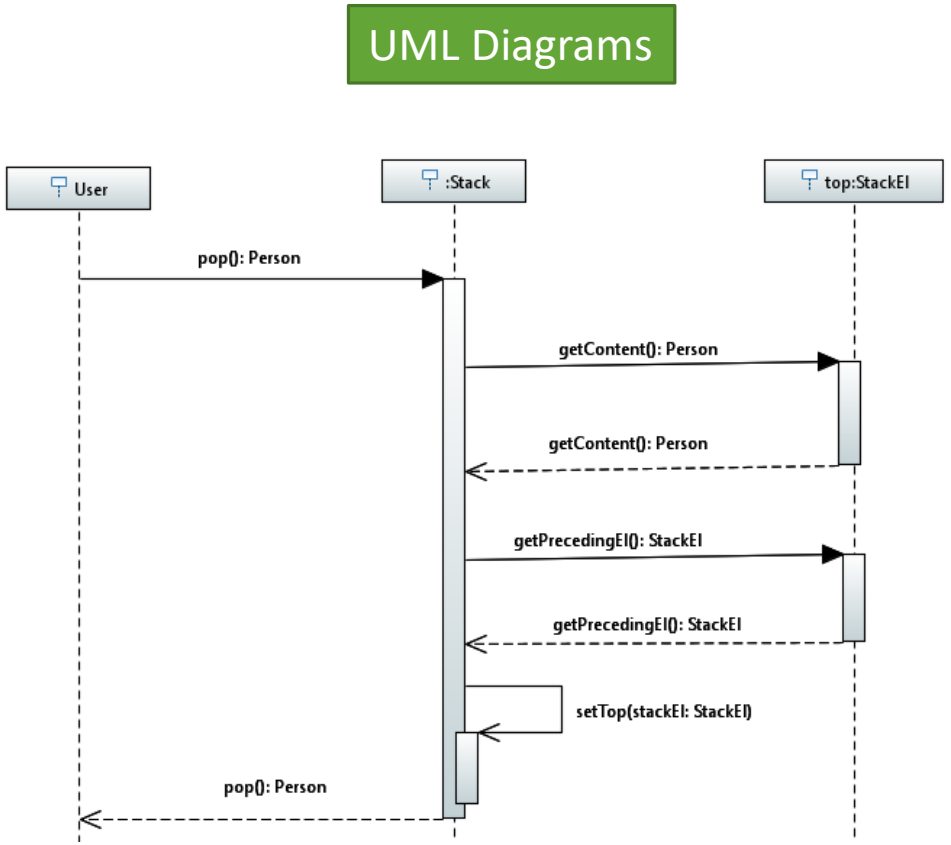


Ejemplo

UML diagrams used



PROV-Template Approach



Ejemplo

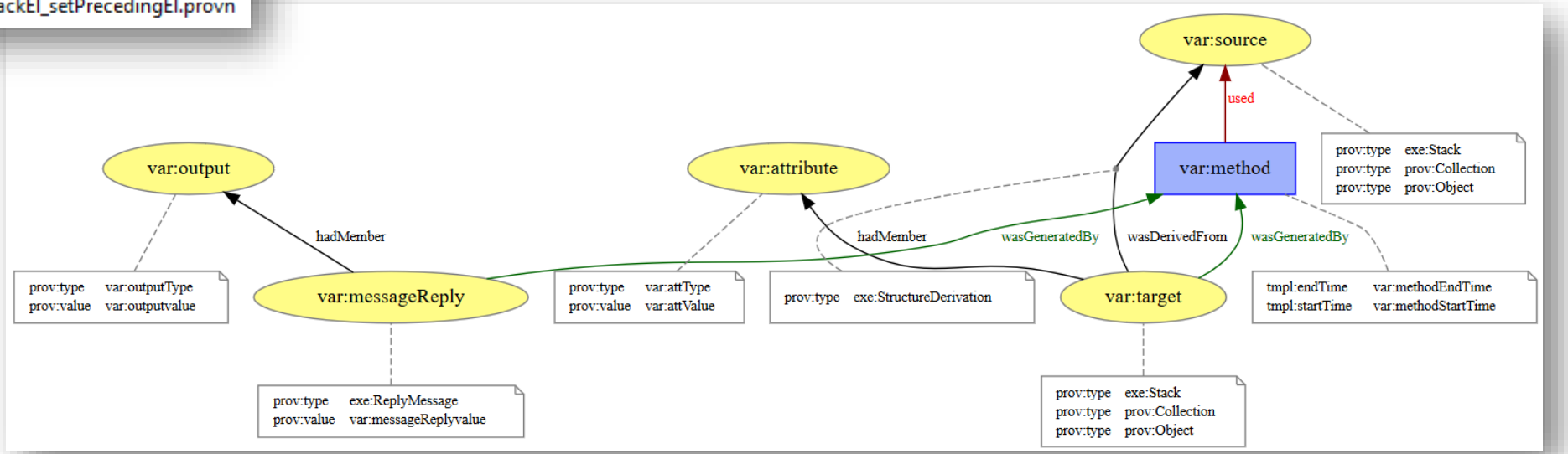
PROV-Templates generated

PROV templates
(UML class diagram)

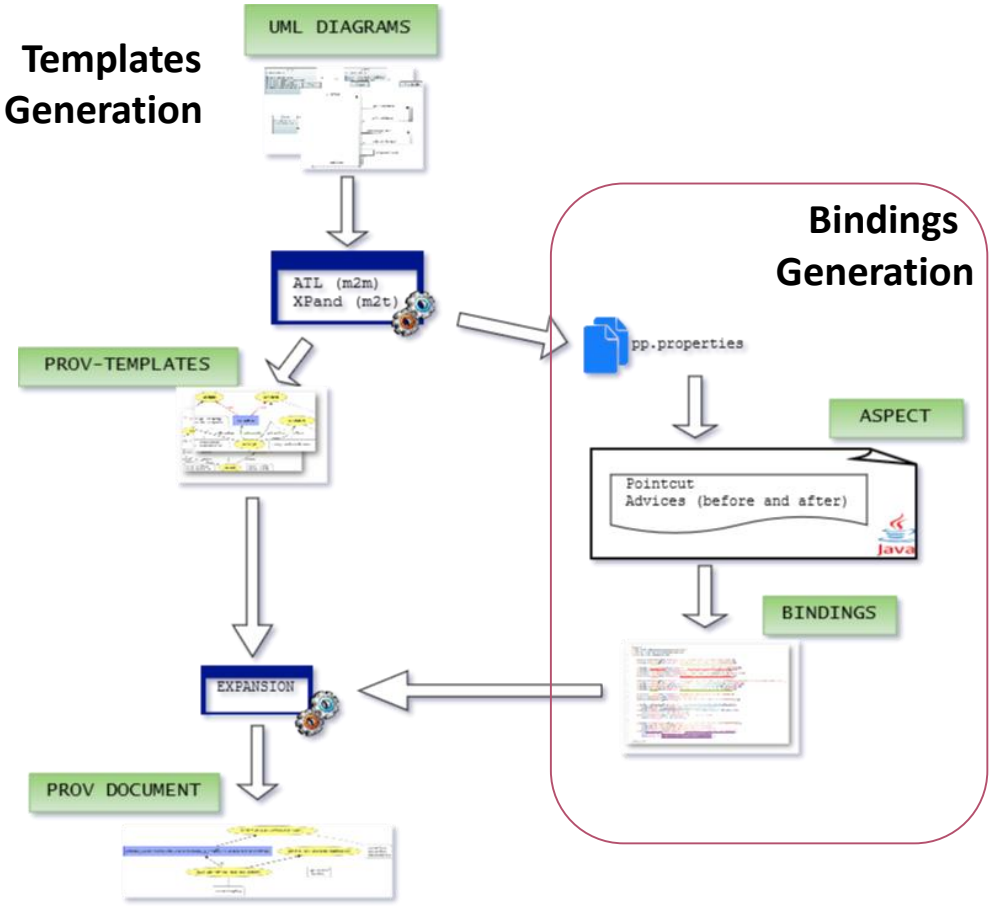
- Person_getLastName.provn
- Person_getName.provn
- Person_setLastName.provn
- Person_setName.provn
- Person_sourceStructure.provn
- Stack_ahio.provn
- Stack_empty.provn
- Stack_getSize.provn
- Stack_pop.provn
- Stack_push.provn
- Stack_setTop.provn
- Stack_sourceStructure.provn
- StackEl_getContent.provn
- StackEl_getPrecedingEl.provn
- StackEl_setContent.provn
- StackEl_setPrecedingEl.provn
- StackEl_sourceStructure.provn

PROV templates
(UML sequence diagram)

- Stack_pop.provn
- Stack_push.provn
- Stack_setTop.provn
- StackEl_getContent.provn
- StackEl_getPrecedingEl.provn
- StackEl_setContent.provn
- StackEl_setPrecedingEl.provn



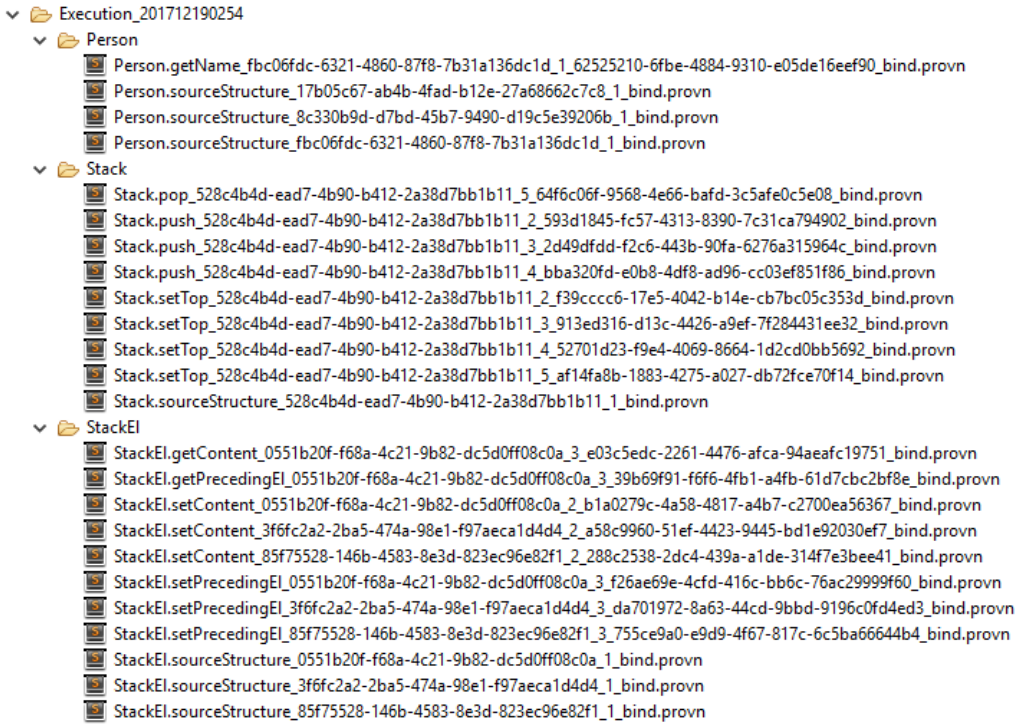
Ejemplo



Application

```java
public class Principal {

    public static void main(String[] args) {

        Stack st = new Stack();
        st.push(new Person("Margaret","Hamilton"));
        st.push(new Person("Gadea","Mata"));
        st.push(new Person("Carol","Shaw"));
        System.out.println(st.pop().getName());

    }
}
```

Ejemplo




```
document
    prefix tmpl <http://openprovenance.org/tmpl#>
    prefix var <http://openprovenance.org/var#>
    prefix exe <http://example.org/>

    entity(var:starter,[tmpl:value_0 = 'exe:4b2ab8ea-bc91-410a-bc13-c999696be1c2'])
    entity(var:source,[tmpl:value_0 = 'exe:528c4b4d-ead7-4b90-b412-2a38d7bb1b11_4'])
    entity(var:messageReply,[tmpl:value_0 = 'exe:42387faf-f901-4329-bf26-c02fb3b926ac'])
    entity(var:outputType,[tmpl:2dvalue_0_0 = 'exe:RootElement.Person@73f792cf'])
    entity(var:outputValue,[tmpl:2dvalue_0_0 = "RootElement.Person@73f792cf" %% xsd:string])
    entity(var:output,[tmpl:value_0 = 'exe:fbc06fdc-6321-4860-87f8-7b31a136dc1d_1'])
    entity(var:method,[tmpl:value_0 = 'exe:pop_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_4_90904229-547a-4a29-b015-9d995c1065fb'])
    entity(var:methodStartTime,[tmpl:2dvalue_0_0 = "2017-12-19T13:02:54" %% xsd:dateTime])
    entity(var:methodEndTime,[tmpl:2dvalue_0_0 = "2017-12-19T13:02:54" %% xsd:dateTime])
    entity(var:message,[tmpl:value_0 = 'exe:pop_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_4_90904229-547a-4a29-b015-9d995c1065fb'])
    entity(var:messageStartTime,[tmpl:2dvalue_0_0 = "2017-12-19T13:02:54" %% xsd:dateTime])
    entity(var:messageEndTime,[tmpl:2dvalue_0_0 = "2017-12-19T13:02:54" %% xsd:dateTime])
    entity(var:event,[tmpl:value_0 = 'exe:pop_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_4_90904229-547a-4a29-b015-9d995c1065fb'])
    entity(var:eventStartTime,[tmpl:2dvalue_0_0 = "2017-12-19T13:02:54" %% xsd:dateTime])
    entity(var:eventEndTime,[tmpl:2dvalue_0_0 = "2017-12-19T13:02:54" %% xsd:dateTime])
    entity(var:target,[tmpl:value_0 = 'exe:528c4b4d-ead7-4b90-b412-2a38d7bb1b11_5'])
    entity(var:object,[tmpl:value_0 = 'exe:528c4b4d-ead7-4b90-b412-2a38d7bb1b11'])
    entity(var:lifeline,[tmpl:value_0 = 'exe:Principal'])
    entity(var:objectSMD,[tmpl:value_0 = 'exe:528c4b4d-ead7-4b90-b412-2a38d7bb1b11_0'])
    entity(var:attribute,[tmpl:value_0 = 'exe:8eee2f7d-efe2-4375-b6eb-0d0f7099a919', tmpl:value_1 = 'exe:c1a6cbe9-6a3d-4c75-b678-6e166273f538'])
    entity(var:attType,[tmpl:2dvalue_0_0 = "class StackExample.StackEl" %% xsd:string, tmpl:2dvalue_1_0 = "int" %% xsd:string])
    entity(var:attValue,[tmpl:2dvalue_0_0 = "StackExample.StackEl@79698539" %% xsd:string, tmpl:2dvalue_1_0 = "2" %% xsd:string])
    entity(var:replyMessages,[tmpl:value_0 = 'exe:bdf7210f-8ce0-467c-aab5-6bcdcce3a25e',
        tmpl:value_1 = 'exe:7a30bdd9-6789-4f59-b9b4-3546ffff4ca0',
        tmpl:value_2 = 'exe:48793ec7-0ba2-41cb-a0fb-e844f3693d9e'])
    entity(var:newRequest,[tmpl:value_0 = 'exe:92044869-b40b-4caa-b302-508518e66828',
        tmpl:value_1 = 'exe:179a7758-1e0c-42d1-839f-d9a444ffecb9',
        tmpl:value_2 = 'exe:8f56af2c-072e-4fc0-b6f8-3c9507bb5dd7'])
endDocument
```

Ejemplo

PROV templates
(UML class diagram)

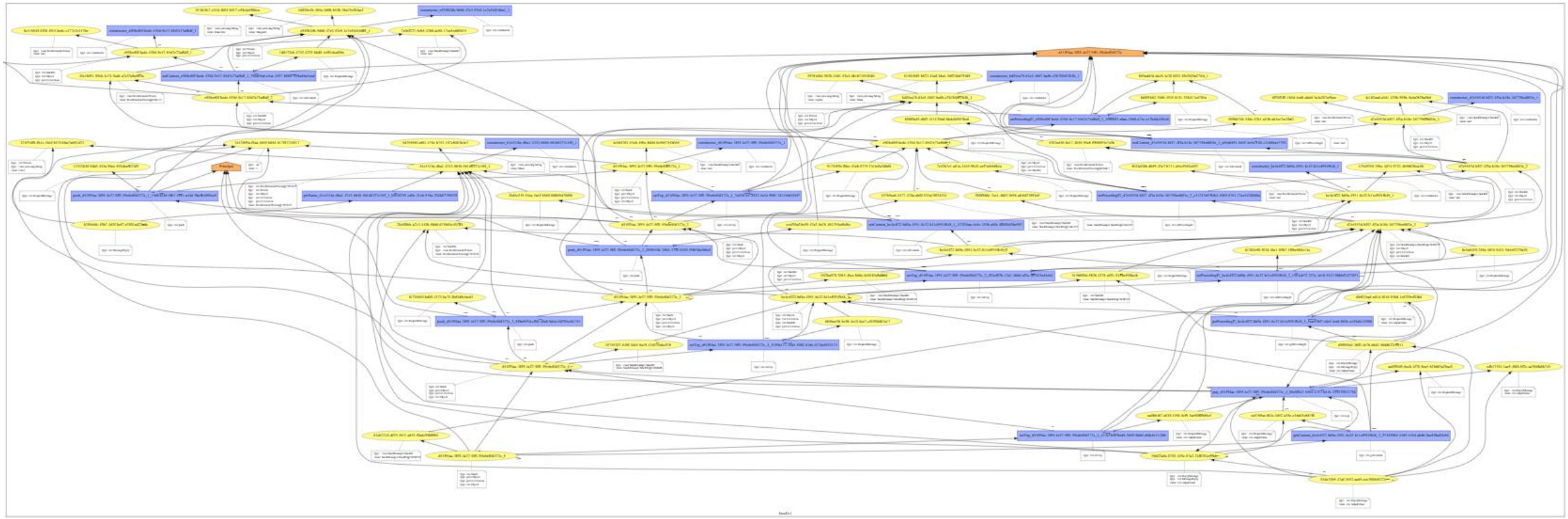
- Person_getLastName.provn
- Person_getName.provn
- Person_setLastName.provn
- Person_setName.provn
- Person_sourceStructure.provn
- Stack_ahio.provn
- Stack_empty.provn
- Stack_getSize.provn
- Stack_pop.provn
- Stack_push.provn
- Stack_setTop.provn
- Stack_sourceStructure.provn
- StackEl_getContent.provn
- StackEl_getPrecedingEl.provn
- StackEl_setContent.provn
- StackEl_setPrecedingEl.provn
- StackEl_sourceStructure.provn

PROV templates
(UML sequence diagram)

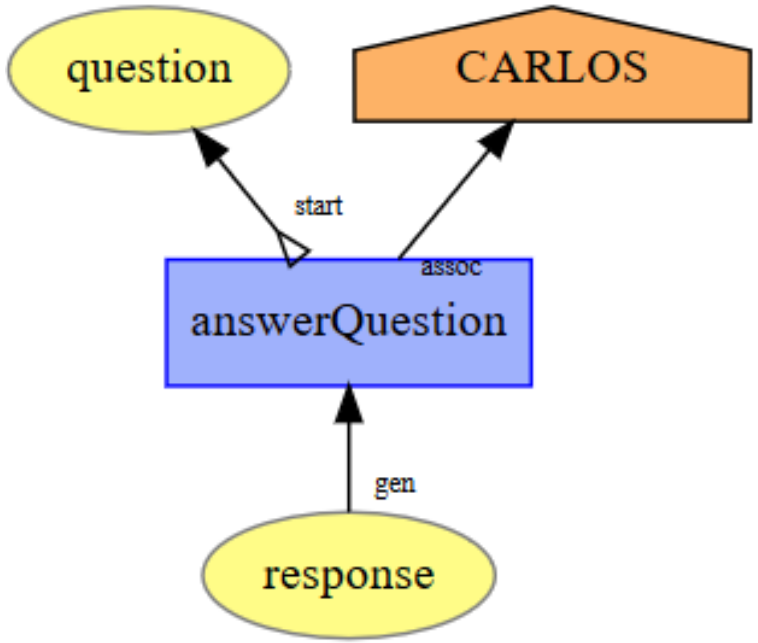
- Stack_pop.provn
- Stack_push.provn
- Stack_setTop.provn
- StackEl_getContent.provn
- StackEl_getPrecedingEl.provn
- StackEl_setContent.provn
- StackEl_setPrecedingEl.provn

- Execution_201712190254
  - Person
    - Person.getName_fbc06fdc-6321-4860-87f8-7b31a136dc1d_1_62525210-6fbe-4884-9310-e05de16eef90_bind.provn
    - Person.sourceStructure_17b05c67-ab4b-4fad-b12e-27a68662c7c8_1_bind.provn
    - Person.sourceStructure_8c330b9d-d7bd-45b7-9490-d19c5e39206b_1_bind.provn
    - Person.sourceStructure_fbc06fdc-6321-4860-87f8-7b31a136dc1d_1_bind.provn
  - Stack
    - Stack.pop_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_5_64f6c06f-9568-4e66-bafd-3c5afe0c5e08_bind.provn
    - Stack.push_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_2_593d1845-fc57-4313-8390-7c31ca794902_bind.provn
    - Stack.push_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_3_2d49dfdd-f2c6-443b-90fa-6276a315964c_bind.provn
    - Stack.push_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_4_bba320fd-e0b8-4df8-ad96-cc03ef851f86_bind.provn
    - Stack.setTop_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_2_f39cccc6-17e5-4042-b14e-cb7bc05c353d_bind.provn
    - Stack.setTop_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_3_913ed316-d13c-4426-a9ef-7f284431ee32_bind.provn
    - Stack.setTop_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_4_52701d23-f9e4-4069-8664-1d2cd0bb5692_bind.provn
    - Stack.setTop_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_5_af14fa8b-1883-4275-a027-db72fce70f14_bind.provn
    - Stack.sourceStructure_528c4b4d-ead7-4b90-b412-2a38d7bb1b11_1_bind.provn
  - StackEl
    - StackEl.getContent_0551b20f-f68a-4c21-9b82-dc5d0ff08c0a_3_e03c5edc-2261-4476-afca-94aeafc19751_bind.provn
    - StackEl.getPrecedingEl_0551b20f-f68a-4c21-9b82-dc5d0ff08c0a_3_39b69f91-f6f6-4fb1-a4fb-61d7cbc2bf8e_bind.provn
    - StackEl.setContent_0551b20f-f68a-4c21-9b82-dc5d0ff08c0a_2_b1a0279c-4a58-4817-a4b7-c2700ea56367_bind.provn
    - StackEl.setContent_3f6fc2a2-2ba5-474a-98e1-f97aeca1d4d4_2_a58c9960-51ef-4423-9445-bd1e92030ef7_bind.provn
    - StackEl.setContent_85f75528-146b-4583-8e3d-823ec96e82f1_2_288c2538-2dc4-439a-a1de-314f7e3bee41_bind.provn
    - StackEl.setPrecedingEl_0551b20f-f68a-4c21-9b82-dc5d0ff08c0a_3_f26ae69e-4cfd-416c-bb6c-76ac29999f60_bind.provn
    - StackEl.setPrecedingEl_3f6fc2a2-2ba5-474a-98e1-f97aeca1d4d4_3_da701972-8a63-44cd-9bbd-9196c0fd4ed3_bind.provn
    - StackEl.setPrecedingEl_85f75528-146b-4583-8e3d-823ec96e82f1_3_755ce9a0-e9d9-4f67-817c-6c5ba66644b4_bind.provn
    - StackEl.sourceStructure_0551b20f-f68a-4c21-9b82-dc5d0ff08c0a_1_bind.provn
    - StackEl.sourceStructure_3f6fc2a2-2ba5-474a-98e1-f97aeca1d4d4_1_bind.provn
    - StackEl.sourceStructure_85f75528-146b-4583-8e3d-823ec96e82f1_1_bind.provn

Ejemplo



Preguntas, discusión, sugerencias, trabajo futuro…



Ejemplo

Luc Moreau https://lucmoreau.wordpress.com/

Dong Huynh http://trungdong.github.io/

Provenance

The Foundations for Provenance on the Web. Luc Moreau
Lineage Retrieval for Scientific Data Processing. A survey. R Bose, J Frew
Provenance and scientific workflows: challenges and opportunities. SB Davidson, J Freire

PROV

PROV-OVERVIEW. Descripción general de la familia de documentos PROV.
PROV-PRIMER. Manual básico del modelo de datos PROV.
PROV-DM. El modelo de datos PROV para provenance.
Graphical Convention to express PROV. https://www.w3.org/2011/prov/wiki/Diagrams
Provenance: An Introduction to PROV. Luc Moreau, Paul Groth
The rationale of PROV. Luc Moreau, Paul Groth , James Cheney, Timothy Lebo, Simon Miles