

# Aplicación de la prueba de mutaciones a composiciones de servicios web en WS-BPEL para la generación de conjuntos de casos de prueba de calidad

Antonia Estero Botaro  
Directores: Inmaculada Medina Bulo  
Juan José Domínguez Jiménez



Departamento de Ingeniería Informática  
Universidad de Cádiz

- 1 Introducción
  - Objetivos y aportaciones
  - El lenguaje WS-BPEL
- 2 Operadores de mutación para WS-BPEL 2.0
- 3 Herramienta MuBPEL
- 4 Formalización de la prueba de mutaciones
- 5 Evaluación y optimización de los operadores de mutación
- 6 Generación de casos de prueba para WS-BPEL
- 7 Publicaciones

# Motivación y Objetivos

## Objetivo

Aplicación de la prueba de mutaciones a las composiciones de servicios web en WS-BPEL 2.0 para definir un método de obtención de casos de prueba de calidad basado en algoritmos genéticos.

## Aportaciones

- 1 Definición de un conjunto de operadores de mutación adaptados a las características del lenguaje WS-BPEL 2.0.
- 2 Implementación de una herramienta de generación y ejecución de mutantes WS-BPEL: MuBPEL.
- 3 Formalización de una parte importante de la prueba de mutaciones e introducción de métricas que nos permiten evaluar los mutantes y los operadores de mutación.
- 4 Evaluación cuantitativa de los operadores de mutación definidos para la optimización de su comportamiento.
- 5 Definición de un procedimiento de generación de casos de prueba basado en la prueba de mutaciones y en algoritmos genéticos.
- 6 Implementación del generador de casos de prueba y obtención de los primeros resultados.

# El lenguaje WS-BPEL

## ¿Qué es WS-BPEL?

Lenguaje basado en XML que permite especificar el comportamiento de un proceso de negocio basado en las interacciones entre servicios web.

## Ejemplo

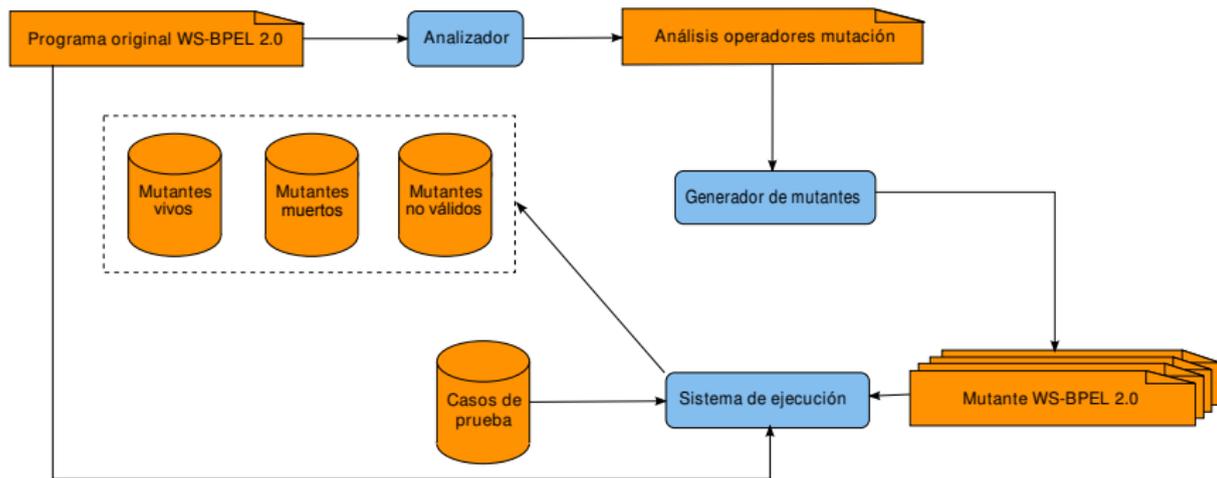
```
<flow> ← Actividad estructurada
  <links> ← Contenedor
    <link name="comprobVuelo-A-reservVuelo" ← Atributo/> ← Elemento
  </links>
  <invoke name="comprobVuelo" ... > ← Actividad básica
    <sources> ← Contenedor
      <source linkName="comprobVuelo-A-reservVuelo" ← Atributo/> ← Elemento
    </sources>
  </invoke>
  <invoke name="comprobHotel" ... />
  <invoke name="comprobAlquilerCoche" ... />
  <invoke name="reservVuelo" ... >
    <targets> ← Contenedor
      <target linkName="comprobVuelo-A-reservVuelo" /> ← Elemento
    </targets>
  </invoke>
</flow>
```

# Operadores de mutación para WS-BPEL 2.0

- Se ha definido un conjunto de 34 operadores. Estos operadores han sido clasificados en 5 categorías:
  - 1 Operadores de reemplazamiento de identificadores (I)
  - 2 Operadores de expresiones (E)
  - 3 Operadores de actividades (A)
  - 4 Operadores de condiciones excepcionales y eventos (X)
  - 5 Operadores de cobertura (C)

# Herramienta MuBPEL

- MuBPEL permite aplicar mutación firme a las composiciones WS-BPEL.



# Formalización de la prueba de mutaciones

Se ha formalizado una parte importante de la teoría de la prueba mutaciones lo que nos ha permitido introducir las siguientes definiciones:

- **Mutante débil** Mutante que es matado por todos los casos de prueba.

$$W = \{m \in M \mid K_m = T\}$$

- **Mutante resistente** Mutante que es matado por un único caso de prueba.

$$R = \{m \in M \mid |K_m| = 1\}$$

- **Calidad de un mutante**

$$Q_m = \begin{cases} 0, & m \in E \\ 1 - \frac{1}{(|M| - |E|) \cdot |T|} \sum_{t \in K_m} |C_t|, & m \in D \end{cases}$$

- **Calidad de un operador de mutación**

$$Q_O = \frac{1}{|M_O|} \sum_{m \in M_O} Q_m = Q_{M_O}$$

# Evaluación y optimización de los operadores de mutación

## Objetivos

- 1 Determinar si todos los operadores de mutación de WS-BPEL son válidos o si podemos eliminar alguno, por ejemplo, porque solo produce mutantes no válidos.
- 2 Determinar si tenemos que modificar la definición o la implementación de algunos operadores, por ejemplo, porque produzca muchos mutantes no válidos o equivalentes.
- 3 Determinar si algunos operadores de mutación son más interesantes que otros en función de su calidad.

## Variables medidas

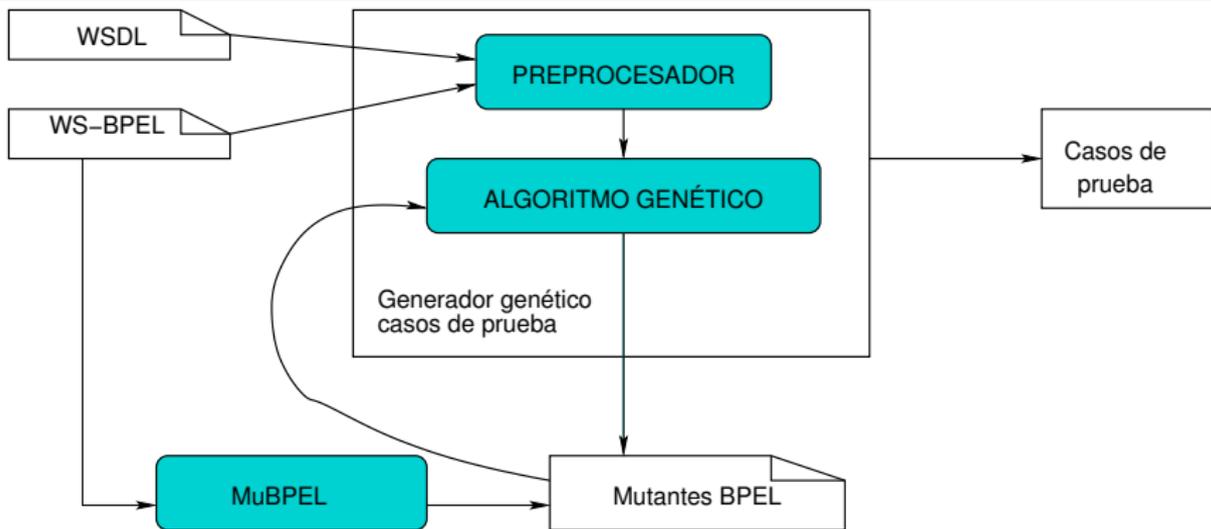
- 1 Número de mutantes equivalentes.
- 2 Número de mutantes no válidos.
- 3 Calidad del operador de mutación.
- 4 Porcentaje de mutantes resistentes
- 5 Porcentaje de mutantes débiles

# Conclusiones de la evaluación

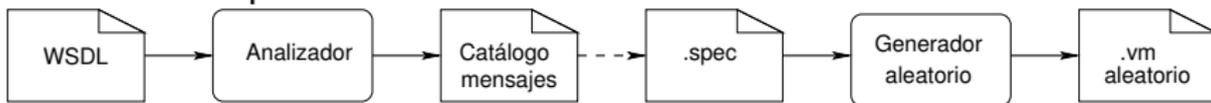
- ACI** Se ha descartado este operador tras comprobar que es imposible escribir una composición para la que pueda producir mutantes válidos.
- EMF** Se ha modificado su definición e implementación para reducir el número de mutantes equivalentes que producían del 50% al 0%.
- ASF** Se ha modificado su implementación para reducir el número de mutantes no válidos que producía del 16.2% al 0%.

Dado que las métricas de calidad definidas dependen de los mutantes y de los conjuntos de casos de prueba utilizados para medirlas. Para asegurar que los valores obtenidos para las distintas composiciones y operadores con comparables, se han utilizado conjuntos de casos de prueba **adecuados**, **no redundantes** y **mínimos**.

# Generación de casos de prueba para WS-BPEL



## Pasos del Preprocesador



# Algoritmo Genético para la generación de casos de prueba

La generación de casos de prueba se realiza mediante un AG, en el que cada individuo representa a un caso de prueba.

## Codificación de un individuo

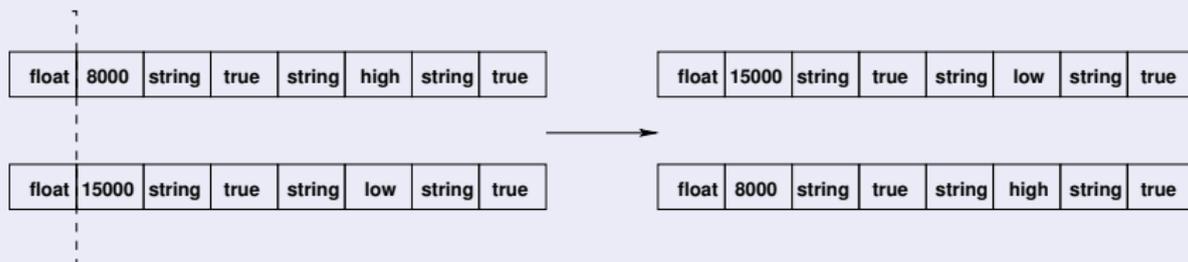
<b>Tipo 1</b>	<b>Valor 1</b>	<b>Tipo 2</b>	<b>Valor 2</b>	...	<b>Tipo N</b>	<b>Valor N</b>			
Lista	Nº elem.	Tipo 1	Valor 1	...	Tupla	Nº elem.	Tipo 1	Valor 1	...

## Aptitud de los individuos

$$Aptitud(X) = \sum_{j=1}^M \frac{e_{Xj}}{\sum_{i=1}^T e_{ij}}$$

# Operadores de cruce y mutación del AG

## Operador de cruce



## Operador de mutación

La mutación se realiza sobre los campos de valor y el cambio a realizar depende del tipo de dato.

## Terminación

- Número máximo de generaciones
- Porcentaje de mutantes muertos
- Estancamiento de la evolución de la aptitud máxima
- Estancamiento de la evolución de la aptitud media

# Publicaciones

- Estero Botaro, A.; Palomo Lozano, F. y Medina Buló, I.  
*Operadores de mutación para WS-BPEL 2.0.*  
Actas de "Talleres de Ingeniería del Software y Bases de Datos" (PRIS08). XIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2008). Gijón. 2008.
- Estero Botaro, A.; Palomo Lozano, F. y Medina Buló, I.  
*Mutation operators for WS-BPEL 2.0.*  
Proceedings of 21st edition of the International Conference on Software & Systems Engineering and their Applications (ICSSEA 2008). París, Francia. 2008.
- Estero Botaro, A.; Domínguez Jiménez, J.J. y Medina Buló, I.  
*Una arquitectura para la generación de casos de prueba de composiciones WS-BPEL basada en mutaciones.*  
Actas de "Talleres de Ingeniería del Software y Bases de Datos 2009" (PRIS09). XV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2009). San Sebastián. 2009.

- Medina Buló, I.; Domínguez Jiménez, J.J. y Estero Botaro, A.  
*Un sistema para la generación automática de mutantes de composiciones WS-BPEL.*  
Actas de las V Jornadas Científico-Técnicas en Servicios WEB y SOA (JSWEB 2009). Madrid. 2009.
- Estero Botaro, A.; Palomo Lozano, F. y Medina Buló, I.  
*Quantitative Evaluation of Mutation Operators for WS-BPEL Compositions.*  
Proceedings of the 5th International Workshop on Mutation Analysis (Mutation 2010). París. 2010.
- Domínguez Jiménez, J.J.; Estero Botaro, A.; García Domínguez, A. y Medina Buló, I. *Evolutionary Mutation Testing.* Information and Software Technology. Vol. 53, N<sup>o</sup> 10, pág. 1108–1123, Octubre 2011.

- Estero Botaro, A.; Domínguez Jiménez, J.J.; Gutiérrez Madroñal, L. y Medina Buló, I.  
*Evaluación de la calidad de los mutantes en la prueba de mutaciones.*  
Actas de las XV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2011). La Coruña. 2011.
- Estero Botaro, A.; Boubeta Puig J.; Liñeiro V.; Medina Buló, I.  
*Operadores de mutación de cobertura para WS-BPEL 2.0*  
Actas de las XVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2012). Almería. 2012.
- García Domínguez, A.; Estero Botaro, A.; Medina Buló, I. y Palomo Lozano, F.  
*MuBPEL: Una herramienta de mutación firme para WS-BPEL 2.0.*  
Actas de las XVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2012). Almería. 2012.

- Estero Botaro, A.; Palomo-Lozano, F.; Medina-Bulo, I.; Domínguez-Jiménez, J.J. y García-Domínguez, A.  
*Quality metrics for mutation testing*  
Enviado al número especial sobre Mutation Testing de la revista Software Testing, Verification and Reliability.
- *Genetic Test Case Generation for Improving the Quality of Test Suites for WS-BPEL Compositions* En preparación para enviar a ISSTA 2013.