

Capítulo 1

DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

1.1. Tecnologías y herramientas empleadas

En esta sección se enumeran las tecnologías y herramientas usadas a lo largo del proyecto.

- Análisis y modelado de la aplicación
 - Visual Paradigm 12.2
- Lenguaje de programación y frameworks
 - Ya que uno de los requisitos principales del proyecto es el uso del librería de la doctora K.Bertet implementada en Java, tanto las herramientas como los algoritmos, se han implementado en este lenguaje de programación en su versión 8
 - JavaFX 2 como framework para la interfaz de usuario.
 - JAXB 2.0 para la persistencia de información en archivos como registro de benchmarks, algoritmos y resultados.
 - JUnit 4.10 para tests unitarios.
- Herramientas de desarrollo
 - Netbeans 8.0.2.
 - Maven 3.
 - JavaFX Scene Builder 2.0 para el diseño de la interfaz de usuario.
- Documentación
 - Esta memoria ha sido redactada con Latex y Microsoft Word 2013.

1.2. Casos de uso

En 1986, Ivar Jacobson, importante contribuyente al desarrollo de los modelos de UML y proceso unificado, creó el concepto de caso de uso.

Durante los años 1990 los casos de uso se convirtieron en una de las prácticas más comunes para la captura de requisitos funcionales, especialmente con el desarrollo del paradigma de la programación orientada a objetos, donde se originaron, si bien puede utilizarse con resultados igualmente satisfactorios con otros paradigmas de programación.

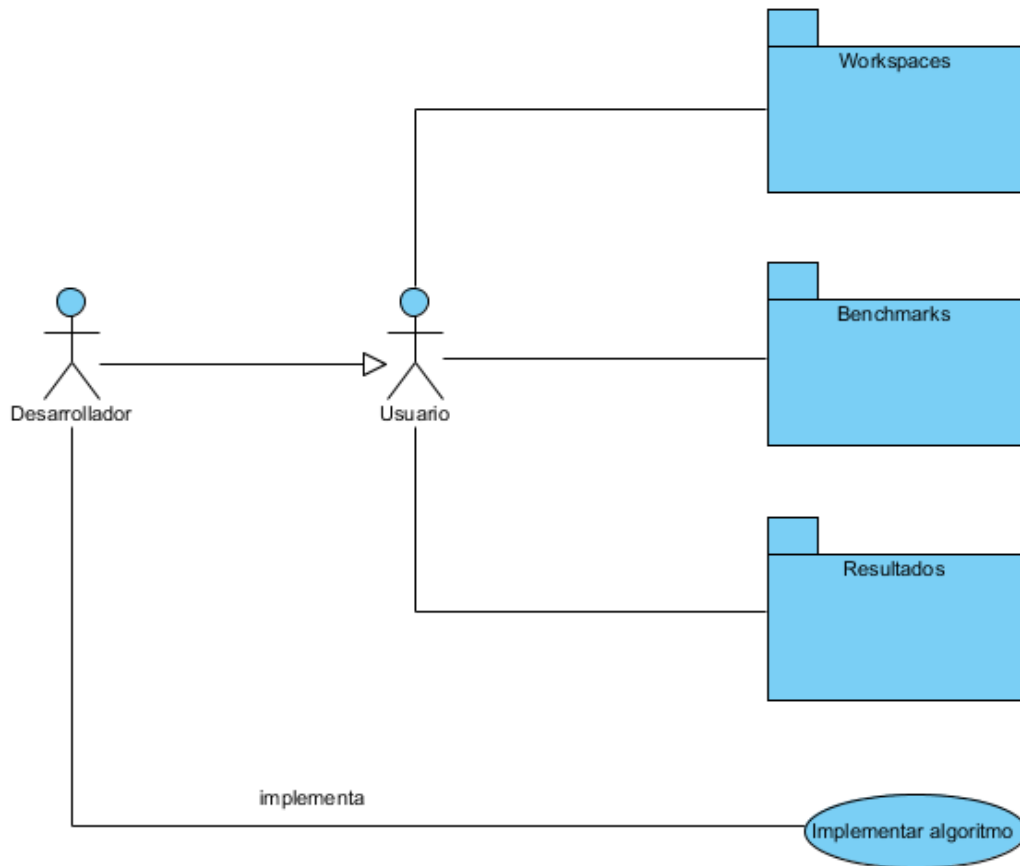
Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio







sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.

La aplicación principal de los casos de uso es en el proceso de análisis y diseño pero de manera particular en la definición de requerimientos del usuario. Es una excelente herramienta de comunicación debido a la sencillez de su elaboración así como su comprensión.

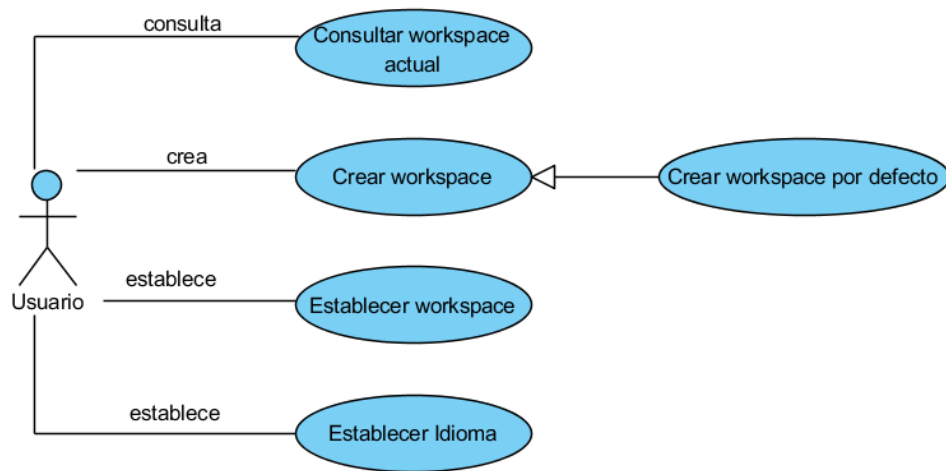
En esta sección, se presentan los diagramas de casos de uso diseñados para el análisis y diseño de la herramienta *IS Bench* y sus especificaciones.

1.2.1. Inicio



Name	Description
 Desarrollador	Implementa los algoritmos que el sistema ejecuta.
 Usuario	Usuario general de la aplicación. No existirán roles ni permisos especiales. Cualquier usuario puede realizar cualquier acción.
 Implementar algoritmo	Un desarrollador puede implementar algoritmos para que el sistema los ejecute. Éstos han de ser implementaciones Java que deben cumplir las siguientes condiciones: <ol style="list-style-type: none"> Han de encontrarse en la ruta que se indique en la propiedad <i>algorithms.classpath</i> en las preferencias del <i>workspace</i> actual. Por defecto, es la carpeta <i>lib</i> de dicho <i>workspace</i>. Debe implementar la interfaz <i>es.uma.pfc.is.algorithms.Algorithm</i>.
 Workspaces	Gestión de workspaces.
 Resultados	Consulta de resultados.
 Benchmarks	Registro y ejecución de benchmarks y algoritmos.

1.2.2. Workspaces



Summary

Name	Description
Usuario	Usuario general de la aplicación.
Crear workspace por defecto	Creación del workspace inicial en el primer arranque de la aplicación.
Consultar workspace actual	Consultar información del workspace actual.
Establecer Idioma	<p>Es posible mostrar la aplicación en distintos idiomas, siendo así más amigable para el usuario.</p> <p>El idioma por defecto, será el establecido en la máquina.</p> <p>Para cambiar el idioma de la aplicación, se deberá acceder a la pestaña "Language" de la ventana "Preferences", que se puede abrir desde la opción de menú "Preferences".</p> <p>Los idiomas disponibles inicialmente son:</p> <ul style="list-style-type: none"> • Español • Inglés
Establecer workspace	Cambio de workspace.
Crear workspace	Crear un workspace.

Description

Para la comodidad de uso de la herramienta, se implementará un sistema de configuración de usuario de forma que éste pueda centralizar las entradas y salidas en un directorio de trabajo, dar la opción de recordar la última ubicación de la cual se ha seleccionado una entrada o una salida, organizar conjuntos de pruebas, etc.

Details



Usuario

Name	Value
Description	Usuario general de la aplicación. No existirán roles ni permisos especiales. Cualquier usuario puede realizar cualquier acción.
ID	AC01
Visibility	public



Crear workspace por defecto

Name	Value
Description	<p>La primera vez que se ejecuta la aplicación, se crea automáticamente:</p> <ul style="list-style-type: none">• El archivo de configuración general \.isbench\isbench.properties• El workspace por defecto con:<ul style="list-style-type: none">○ El directorio \.isbench\default, \.isbench\default\input y \.isbench\default\output.○ El archivo \.isbench\default\preferences.properties con las preferencias por defecto.• Se registra el workspace creado anteriormente en el archivo de configuración general \.isbench\isbench.properties con la propiedad: workspace.default=\.isbench\default• Se establece como workspace actual, el workspace por defecto estableciendo la propiedad workspace.current en el archivo El archivo de configuración general \.isbench\isbench.properties :

Name	Value
	workspace.current= workspace.default
ID	UC26
Stereotypes	UseCase
Requirements	Crear workspace por defecto

Crear workspace por defecto en la primera ejecución

1. Ejecutar por primera vez la aplicación.
2. **SYSTEM** Se crea los directorios \.isbench\default, \.isbench\default\input y \.isbench\default\output.
3. **SYSTEM** Se crea el archivo \.isbench\default\preference.properties.
4. **SYSTEM** El sistema crea el archivo \.isbench\isbench.properties.
5. **SYSTEM** Se establece como workspace actual el workspace creado, estableciendo la propiedad `workspace.current=\\.isbench\default`

Consultar workspace actual

Name	Value
Description	Consultar información del workspace actual.
ID	UC24
Stereotypes	UseCase
Requirements	Gestor de configuración general de la aplicación, Implementación de ventana "Workspaces"

Primer acceso al sistema

1. El usuario accede al sistema.
2. Selecciona la opción de menú "Preferencias -> Workspaces".
3. **SYSTEM** Se crea el archivo \.isbench\isbench.properties.
4. **SYSTEM** Se crea un workspace por defecto en el directorio \.isbench\default y lo registra en el archivo \.isbench\isbench.properties con la propiedad, `workspace.default=\\.isbench\default`
5. **SYSTEM** Se establece como workspace actual el workspace creado, estableciendo la propiedad `workspace.current=\\.isbench\default`
6. **SYSTEM** Se muestra la ventana "Workspaces" con el workspace actual y sus preferencias.

Consultar workspace actual

1. Acceder al menú "Preferencias -> Workspaces"
2. **SYSTEM** Se muestra la ventana "Workspaces" con el workspace seleccionado y las preferencias asociadas a éste.

Establecer Idioma

Name	Value
Description	<p>Es posible mostrar la aplicación en distintos idiomas, siendo así más amigable para el usuario.</p> <p>El idioma por defecto, será el establecido en la máquina.</p> <p>Para cambiar el idioma de la aplicación, se deberá acceder a la pestaña "Language" de la ventana "Preferences", que se puede abrir desde la opción de menú "Preferences".</p> <p>Los idiomas disponibles inicialmente son:</p> <p>Español</p> <ul style="list-style-type: none">• Inglés
ID	UC13
Stereotypes	UseCase

Cambiar el idioma

1. Acceder a la opción de menú "Preferences"
2. Seleccionar la pestaña "Language"
3. Seleccionar un idioma del desplegable.
4. Pulsar el botón "Ok".

Cancelar el cambio de idioma

1. Acceder a la opción de menú "Preferences"
2. Seleccionar la pestaña "Language"
3. Seleccionar un idioma del desplegable.
4. Pulsar el botón "Cancel".

Establecer workspace

Name	Value
Description	<p>El usuario podrá cambiar de workspace desde la ventana "Workspaces" que se abrirá desde la opción de menú "Preferences -> Workspaces".</p> <p>El usuario ha de pulsar el botón "Switch" y el sistema mostrará un desplegable con los workspaces registrados.</p> <p>El usuario ha de seleccionar uno de los items y pulsar "Ok". Seguidamente, el sistema se actualizará con la información del nuevo workspace seleccionado, cargando los benchmarks y resultados registrados en él.</p> <p>También podrá cancelar la acción, pulsando "Cancel".</p>
ID	UC12
Stereotypes	UseCase

Cambiar de workspace

1. Acceder a la opción de menú "Preferences -> Workspaces"
2. Seleccionar del desplegable "Workspaces" un workspace distinto al actual.
3. Seleccionar "Guardar".
4. **SYSTEM** Se crea la propiedad **worspace.change** en `\.isbench\isbench.properties`
5. **SYSTEM** Se muestra un mensaje informando de que el cambio se hará la próxima vez que se arranque la aplicación.

Cancelar el cambio de workspace

1. Acceder a la opción de menú "Preferences -> Workspaces"
2. **SYSTEM** Se muestra la ventana "Workspaces"
3. Seleccionar del desplegable "Workspaces" un workspace distinto al actual.
4. Seleccionar "Cancelar".

Aplicar cambio de workspace

1. Arrancar la aplicación.
2. **SYSTEM** El sistema comprueba que existe la propiedad **worspace.change** en `\.isbench\isbench.properties`.
3. **SYSTEM** Se comprueba que existe el archivo `workspace.xml` en el directorio que indica la

propiedad **workspace.change** .

4. **SYSTEM** Actualiza la propiedad **workspace.default** en el mismo archivo con el valor de **workspace.change** y borra esta última

5. **SYSTEM** Se carga la aplicación con el workspace actualizado.

Aplicar cambio de workspace incorrecto

1. Arrancar la aplicación.

2. **SYSTEM** El sistema comprueba que existe la propiedad **workspace.change** en `\.isbench\isbench.properties`.

3. **SYSTEM** Se comprueba que NO existe el archivo `workspace.xml` en el directorio que indica la propiedad **workspace.change** .

4. **SYSTEM** Muestra un mensaje de error indicando que el workspace actual no es correcto y muestra la ventana Workspaces para que se registre correctamente.

Crear workspace

Name	Value
Description	
	1 Workspace
	<p>Un <i>workspace</i> será una ubicación física donde se guardarán archivos relacionados para su uso en IS Bench.</p> <p>Podrá contener:</p> <ul style="list-style-type: none">• Registro de algoritmos y benchmarks.• Archivos que servirán de entrada a los algoritmos a ejecutar.• Salidas de los algoritmos ejecutados.• Otras preferencias del usuario, como el idioma. <p>Se definirá un <i>workspace</i> por defecto, que podrá ser modificado por el usuario.</p>
	2 Workspace por defecto
	<p>Inicialmente, la aplicación tomará como <i>workspace</i> por defecto la carpeta <code>./isbench/default</code>.</p> <p>Esta carpeta es la localización inicial que abrirán los selectores de entrada y salida y contendrá dos carpetas: <i>inputs</i> y <i>outputs</i>.</p>

Name	Value
	<p>La carpeta <i>inputs</i> será el directorio que se abrirá por defecto al seleccionar la entrada para la ejecución de un algoritmo.</p> <p>La carpeta <i>outputs</i> será el directorio que se abrirá por defecto al seleccionar la salida de la ejecución de un algoritmo.</p> <p>3 Crear un Workspace</p> <p>El usuario podrá crear un workspace accediendo a la ventana "Workspaces" desde la opción de menú "Preferences -> Workspaces".</p> <p>La ventana mostrará un desplegable combinado, en el que sus items serán los workspaces existentes en el archivo [home_usuario]/.isbench/isbench.properties.</p> <p>Podrá crear uno nuevo introduciendo el nombre de un directorio o seleccionándolo con el buscador (botón "...").</p> <p>El sistema creará una carpeta con el nombre introducido que contendrá un archivo workspace.xml. En este archivo se almacenará la configuración del workspace creado.</p> <p>Además el sistema preguntará al usuario si desea establecerlo como workspace actual. Si el usuario confirma, el cambio se hace efectivo la próxima vez que se ejecute la aplicación.</p>
ID	UC11
Stereotypes	UseCase
Justification	Creación de nuevos workspaces
Requirements	Crear un workspace, Implementación de ventana "Workspaces", Carga de workspaces registrados y sus preferencias

Crear un workspace

1. Seleccionar la opción de menú "Preferencias -> Workspaces".
2. **SYSTEM** Se muestra la ventana "Workspaces" con la información del workspace seleccionado actualmente.
3. Seleccionar un directorio con el botón *Examinar (...)*.
4. **SYSTEM** La etiqueta nombre se inicializa con el nombre del directorio y se convierte en editable.
5. Establecer las preferencias deseadas editando la tabla "Preferencias".
6. Seleccionar "Guardar".
7. **SYSTEM** Se crea el directorio seleccionado y en él las carpetas input, output y el archivo preferences.properties.

8. **SYSTEM** Se actualiza el archivo [user_home]\.isbench\isbench.properties añadiendo el nuevo workspace: workspac.nombre=ruta_absoluta_dir

9. **SYSTEM** El sistema pregunta si se desea establecer éste como el workspace actual.

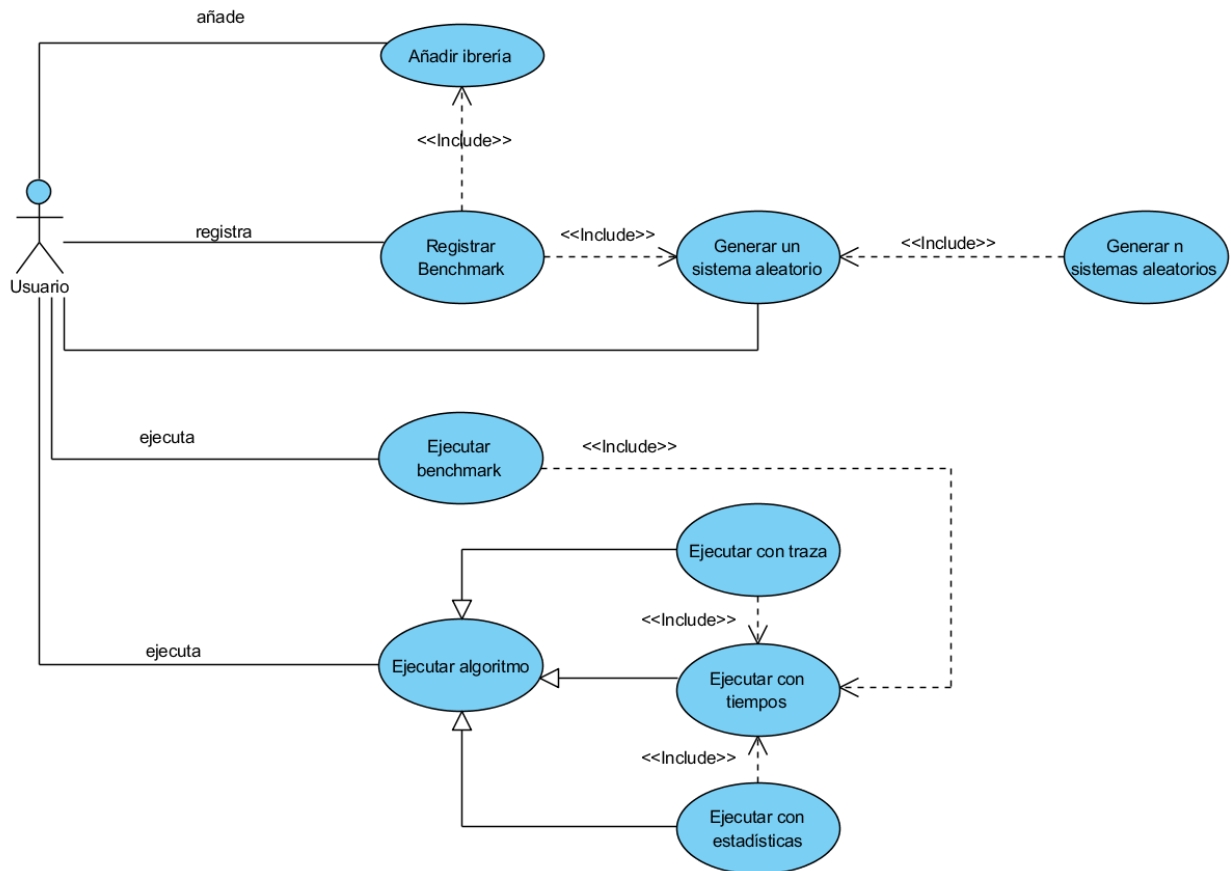
10. **if** se acepta establecerlo como workspace actual

10.1. **SYSTEM** El sistema actualiza el archivo \.isbench\isbench.properties estableciendo la propiedad workspace.change=workspace.nombre

10.2. **SYSTEM** Se muestra un mensaje informando de que el cambio se hará la próxima vez que se arranque la aplicación.


end if

1.2.3. Benchmarks



Summary

Name	Description
Usuario	Usuario general de la aplicación. No existirán roles ni permisos especiales. Cualquier usuario puede realizar cualquier acción.
Añadir librería	Añadir al workspace librerías externas con implementaciones de algoritmos sobre sistemas implicacionales.
Registrar Benchmark	Registro de benchmarks en el workspace.
Generar un sistema aleatorio	Generación de un conjunto de implicaciones aleatorio.
Generar n sistemas aleatorios	Generación de n conjuntos de implicaciones aleatorios.
Ejecutar benchmark	Ejecución de un benchmark.
Ejecutar algoritmo	Ejecución de un algoritmo.
Ejecutar con estadísticas	Ejecución de algoritmos con el modo “Statistics” activo.
Ejecutar con tiempos	Ejecución de algoritmos con el modo “Time” activo.

Name	Description
 Ejecutar con traza	Ejecución de algoritmos con el modo “Trace” activo.

Details

Usuario

Name	Value
Description	<p>Usuario general de la aplicación.</p> <p>No existirán roles ni permisos especiales. Cualquier usuario puede realizar cualquier acción.</p>
ID	AC01
Visibility	Public

Añadir librería

Name	Value
Description	Desde IS Bench se podrán añadir al workspace librerías externas con implementaciones de algoritmos sobre sistemas implicacionales.
ID	UC09
Stereotypes	UseCase
Justification	Registro de algoritmos en el workspace para su ejecución.
Preconditions	Las librerías a añadir contienen implementaciones de algoritmos sobre sistemas implicacionales que cumplen la API definida.

Añadir una librería

1. Hacer click en el botón “+” de la lista de algoritmos
2. Seleccionar uno o varios ficheros JAR y hacer click en Aceptar.
3. SYSTEM: Copia los ficheros seleccionados al workspace actual.
4. SYSTEM: Recarga la lista de algoritmos con los encontrados en las librerías añadidas.


Añadir una librería ya existente

1. Hacer click en el botón “+” de la lista de algoritmos
2. Seleccionar uno o varios ficheros JAR y hacer click en Aceptar.
3. SYSTEM: Pregunta si se desea sobrescribir el fichero.
4. Confirmar la acción.
5. SYSTEM: Copia los ficheros seleccionados al workspace actual.
6. SYSTEM: Recarga la lista de algoritmos con los encontrados en las librerías añadidas.

Cancelar la adición de una librería ya existente

1. Hacer click en el botón “+” de la lista de algoritmos
2. Seleccionar uno o varios ficheros JAR y hacer click en Aceptar.
3. SYSTEM: Pregunta si se desea sobrescribir el fichero.
4. Cancelar la acción.

Registrar Benchmark

Name	Value
Description	<p>En el sistema existirá una pantalla <i>Benchmarks</i> que constará de dos pestañas:</p> <ul style="list-style-type: none"> • Add • Run <p>Para el registro de benchmarks, el usuario deberá acceder a la pestaña "Add".</p> <p>En esta pestaña se mostrará un formulario con los siguientes campos:</p> <ul style="list-style-type: none"> • "Name": Nombre del Benchmark. • "Input": Directorio que contiene los sistemas implicacionales de entrada para todos los algoritmos del Benchmark. <p>El directorio de entrada por defecto es [workspace_actual]/[nombre_benchmark]/input.</p> <p>A continuación de este campo se muestra un botón con dos opciones: <i>Seleccionar directorio</i>, <i>Generar</i>.</p> <p><i>Seleccionar directorio</i>: abre un cuadro de diálogo para seleccionar un directorio con ficheros de entrada.</p> <p><i>Generar</i>: Abre el generador de implicaciones, para crear los ficheros de entrada.</p> <ul style="list-style-type: none"> • Sección "Algorithms": Consta de dos listas de algoritmos. <ul style="list-style-type: none"> ○ A la izquierda se mostrará la lista de algoritmos registrados en el workspace actual. Éstos podrán ser filtrados mediante un filtro en vivo situado en la parte superior de la lista. ○ A la derecha la lista de algoritmos que se incluirán en el benchmark que se está creando. Éstos se seleccionarán con doble click de la lista anterior. <p>Todos los campos serán obligatorios.</p>
ID	UC10
Stereotypes	UseCase
Requirements	Diseño de formulario de registro de benchmarks., Carga de lista de algoritmos registrados en el workspace actual., Selección de sistema aleatorio como entrada., Guardar benchmark.
Preconditions	 Registrar Algoritmo

Registrar un benchmark

1. Acceder a la pantalla "Nuevo Benchmark".
2. Introducir un nombre en el campo "Nombre".
3. Introducir la ruta absoluta de un archivo que contiene un sistema implicacional en el campo "Entrada".
4. Seleccionar al menos un algoritmo de la lista de algoritmos.
5. Seleccionar la opción "Guardar".

Registrar un benchmark con un sistema de entrada aleatorio

1. Acceder a la pantalla "Nuevo Benchmark".
2. Introducir un nombre en el campo "Nombre".
3. Seleccionar la opción "Random" del botón del campo Input.
4. El sistema abre una ventana con el Generador de implicaciones.
5. Introducir número de atributos e implicaciones.
6. Seleccionar "Generate".
7. Seleccionar "Save".
8. Cerrar la ventana del generador.
9. Seleccionar al menos un algoritmo de la lista de algoritmos.
10. Seleccionar la opción "Guardar".

Registrar un benchmark vacío

1. Acceder a la pantalla "Nuevo Benchmark".
2. Seleccionar la opción "Guardar".
3. El sistema muestra un mensaje de error indicando que hay campos vacíos.
4. Introducir un nombre en el campo "Nombre".
5. Introducir la ruta absoluta de un archivo que contiene un sistema implicacional en el campo "Entrada".
6. Seleccionar al menos un algoritmo de la lista de algoritmos.
7. Seleccionar la opción "Guardar".

Registrar un benchmark sin algoritmos

1. Acceder a la pantalla "Nuevo Benchmark".
2. Introducir un nombre en el campo "Nombre".
3. Introducir la ruta absoluta de un archivo que contiene un sistema implicacional en el campo "Entrada".
4. Seleccionar la opción "Guardar".
5. El sistema muestra un mensaje de error indicando que se debe seleccionar al menos un algoritmo.
6. Seleccionar al menos un algoritmo de la lista de algoritmos.

7. Seleccionar la opción "Guardar".

Registra un benchmark si sistema de entrada

1. Acceder a la pantalla "Nuevo Benchmark".
2. Introducir un nombre en el campo "Nombre".
3. Seleccionar al menos un algoritmo de la lista de algoritmos.
4. Seleccionar la opción "Guardar".
5. El sistema muestra un mensaje de error indicando que se el campo "Entrada" no puede ser vacío.
6. Introducir la ruta absoluta de un archivo que contiene un sistema implicacional en el campo "Entrada".
7. Seleccionar la opción "Guardar".

Registra un benchmark sin nombre

1. Acceder a la pantalla "Nuevo Benchmark".
2. Introducir la ruta absoluta de un archivo que contiene un sistema implicacional en el campo "Entrada".
3. Seleccionar al menos un algoritmo de la lista de algoritmos.
4. Seleccionar la opción "Guardar".
5. El sistema muestra un mensaje de error indicando que el campo "Nombre" no puede ser vacío.
6. Introducir un nombre en el campo "Nombre".
7. Seleccionar la opción "Guardar".

Registrar un benchmark con un nombre existente

1. Acceder a la pantalla "Nuevo Benchmark".
2. Introducir el nombre de un benchmark existente en el workspace actual en el campo "Nombre".
3. Introducir la ruta absoluta de un archivo que contiene un sistema implicacional en el campo "Entrada".
4. Seleccionar al menos un algoritmo de la lista de algoritmos.
5. Seleccionar la opción "Guardar".
6. El sistema muestra un mensaje de confirmación del benchmark existente con el mismo nombre.
7. Seleccionar la opción "Ok".

Cancelar el registro de un benchmark con un nombre existente

1. Acceder a la pantalla "Nuevo Benchmark".
2. Introducir el nombre de un benchmark existente en el workspace actual en el campo

"Nombre".

3. Introducir la ruta absoluta de un archivo que contiene un sistema implicacional en el campo "Entrada".

4. Seleccionar al menos un algoritmo de la lista de algoritmos.

5. Seleccionar la opción "Guardar".

6. El sistema muestra un mensaje de confirmación del benchmark existente con el mismo nombre.

7. Seleccionar la opción "Cancelar".

Generar un sistema aleatorio

Name	Value
Description	<p>El sistema permite generar sistemas implicacionales aleatorios, que sirvan de entrada a un algoritmo.</p> <p>Estos sistemas son generados a partir de una serie de parámetros que el usuario debe proporcionar:</p> <ul style="list-style-type: none">• Número de atributos. Obligatorio. Debe ser mayor que 0.• Número de implicaciones. Obligatorio. Debe ser mayor que 0.• Tipo de atributos: Se puede seleccionar entre numéricos (1, 2,3,...), alfabéticos (a, b, c, ...) y alfanuméricos (a1, a2, a3, ...). Por defecto será numérico.• Mínimo y Máximo número de atributos en la premisa. No es obligatorio y por defecto sólo se establece el mínimo a 1 atributo.• Mínimo y Máximo número de atributos en la conclusión. No es obligatorio y por defecto sólo se establece el mínimo a 1 atributo.• Salida: Obligatorio para guardar el sistema en fichero, pero no antes. Será la ruta absoluta del fichero en el que se guardará el sistema generado. <p>Una vez introducidos los parámetros, el usuario podrá generar el sistema y la aplicación previsualizará éste.</p> <p>El usuario podrá guardarlo en la ruta introducida.</p>
ID	UC01

Generar un sistema aleatorio con valores por defecto

1. Introducir el número de atributos.

2. Introducir el número de implicaciones.

3. Pulsar el botón "Generate".

4. **SYSTEM** Se genera un sistema con el número de atributos e implicaciones especificados.
5. **SYSTEM** El tipo de los atributos es numérico, el tamaño mínimo de las premisas y de las conclusiones es 1.

Generación de un sistema aleatorio con 0 atributos.

1. Introducir el número de implicaciones.
2. Pulsar el botón "Generate".
3. **SYSTEM** Se muestra un error indicando que el número de atributos ha de ser mayor que 0.

Generación de un sistema aleatorio con 0 implicaciones.

1. Intorducir el número de atributos.
2. Pulsar el botón "Generate".
3. **SYSTEM** Se muestra un error indicando que el número de implicaciones ha de ser mayor que 0.

Generación de un sistema aleatorio con el tipo de atributos a, b, c,...

1. Introducir el número de atributos.
2. Introducir el número de implicaciones.
3. Seleccionar el tipo a, b, c...
4. Pulsar el botón "Generate".
5. **SYSTEM** Se genera un sistema con el número de atributos e implicaciones especificados y el tipo de los atributos es *a, b, c...*

Generación de un sistema aleatorio acotando el tamaño de la premisa, tanto mínimo como máximo.

1. Introducir el número de atributos.
2. Introducir el número de implicaciones.
3. Introducir un tamaño mínimo y máximo para la premisa.
4. **if** el tamaño mínimo de la premisa es **mayor** que el tamaño máximo
 - 4.1. **SYSTEM** Se muestra un mensaje de error indicando que el rango es incorrecto.
5. **else**
 - 5.1. Pulsar el botón "Generate".
 - 5.2. **SYSTEM** Se genera un sistema con el número de atributos e implicaciones especificados.
 - 5.3. **SYSTEM** El tipo de los atributos es numérico.
 - 5.4. **SYSTEM** La longitud de las premisas está entre el mínimo y máximo introducidos.
- end if**

Generación de un sistema aleatorio acotando el tamaño de la conclusión, tanto mínimo como máximo.

1. Introducir el número de atributos.
 2. Introducir el número de implicaciones.
 3. Introducir un tamaño mínimo y máximo para la **conclusión**.
 4. **if** el tamaño mínimo de la conclusión es **mayor** que el tamaño máximo
 - 4.1. **SYSTEM** Se muestra un mensaje de error indicando que el rango es incorrecto.
 5. **else**
 - 5.1. Pulsar el botón "Generate".
 - 5.2. **SYSTEM** Se genera un sistema con el número de atributos e implicaciones especificados.
 - 5.3. **SYSTEM** El tipo de los atributos es numérico.
 - 5.4. **SYSTEM** La longitud de las c está entre el mínimo y máximo introducidos.
- end if**

Guardar el sistema generado en un archivo.

1. Introducir el número de atributos.
2. Introducir el número de implicaciones.
3. Pulsar el botón "Generate".
4. **SYSTEM** Se genera un sistema con el número de atributos e implicaciones especificados.
5. Introducir la ruta absoluta del fichero en el que se desea guardar el sistema.
6. Pulsar el botón "Save".
7. **SYSTEM** Se crea / actualiza el archivo introducido con el sistema generado.

Generar n sistemas aleatorios

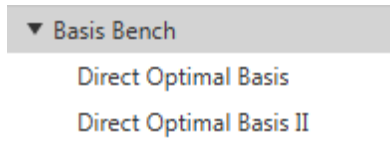
Name	Value
Description	Generación de n conjuntos de implicaciones aleatorios.
ID	UC19
Stereotypes	UseCase



Generar n sistemas aleatorios

1. Introducir el número de atributos.
2. Introducir el número de implicaciones.
3. Introducir un valor mayor que 1 en el número de sistemas a generar.
4. **SYSTEM** El botón "Generate" se deshabilita.

5. Introducir la ruta absoluta del directorio en el que se desea guardar los sistemas generados.
6. Introducir el nombre base de los sistemas.
7. **SYSTEM** El botón "Generate" se habilita.
8. Pulsar el botón "Generate".
9. **SYSTEM** Se genera el número de sistemas introducido, con el número de atributos e implicaciones especificados.
10. **SYSTEM** Se guardan los sigemas en el directorio introducido.
11. **SYSTEM** Los nombres de los archivos es el nombre introducido como nombre base, con el prefijo _1, _2, etc.
12. **SYSTEM** Se muestra un mensaje indicando el número de sistemas generados y dónde se han guardado.

Ejecutar benchmark

Name	Value
Description	<p>En el sistema existirá una pantalla <i>Benchmarks</i> que constará de dos zonas:</p> <ul style="list-style-type: none"> • Añadir • Ejecutar <p>Para la ejecución de benchmarks, el usuario deberá acceder a la zona "Ejecutar".</p> <p>Esta zona se dividirá a su vez en otras dos:</p> <ul style="list-style-type: none"> • A la izquierda se mostrarán los Benchmarks registrados en un árbol de dos niveles, en el que el primer nivel contendrá el nombre del Benchmark y en el segundo nivel, los algoritmos que lo componen. <div style="text-align: center;">  </div> <ul style="list-style-type: none"> • A la derecha se mostrarán los campos de entrada y salida que contendrán las rutas del archivo con el sistema implicacional de entrada y el destino de los resultados de la ejecución correspondientemente. <p>Además, se mostrará un visor de resultados en el que se podrán cargar los archivos generados por la ejecución.</p> <p>El campo "Entrada" se inicializará con el archivo de</p>

Name	Value
	<p>entrada definido para el benchmark seleccionado y el campo "Salida" se inicializará con el directorio //output.</p> <p>Para ejecutar un benchmark el usuario seleccionará uno del árbol y pulsará el botón "Ejecutar".</p> <p>El modo de ejecución que se permite es "Time", generando sólo como salida los sistemas resultantes de cada algoritmo y los tiempos de ejecución de éstos.</p> <p>El sistema ejecutará cada uno de los algoritmos que componen el benchmark seleccionado, generando las salidas en el directorio que se indica en el campo "Salida":</p> <ul style="list-style-type: none"> • Archivos [abreviatura_alg]_output.txt con los sistemas implicacionales resultantes por cada algoritmo ejecutado, , siendo [abreviatura_alg] la abreviatura de cada algoritmo. • Archivos [abreviatura_alg]_history.txt con los tiempos de ejecución por cada algoritmo ejecutado, siendo [abreviatura_alg] la abreviatura de cada algoritmo. • Resumen de la ejecución, contiendo éste para cada algoritmo: <ul style="list-style-type: none"> ○ Algoritmo ejecutado. ○ Fecha y hora. ○ Fichero de salida ○ Tiempo de ejecución. <p>Los datos de este resumen son los que se mostrará cuando el usuario consulte los resultados de las ejecuciones.</p>
ID	UC08
Stereotypes	UseCase
Justification	Ejecución de conjunto de algoritmos para la posterior comparación de sus resultados.
Preconditions	 Registrar Benchmark  Registrar Algoritmo

Ejecución de un benchmark


1. Desde la pantalla principal (Home) hacer click en el botón Benchmarks.
2. Seleccionar un Benchmark en el árbol.
3. Hacer click en el botón "Run".
4. **SYSTEM** Se genera un archivo de salida por algoritmo y otro .log con los tiempos de ejecución.

Ejecución de un benchmark sin tiempos

1. Seleccionar un Benchmark en el árbol.
2. Desactivar la casilla "Time".
3. **SYSTEM** Hacer click en el botón "Run".

Ejecutar algoritmo

Name	Value
Description	<p>El sistema permitirá ejecutar un algoritmo previamente registrado.</p> <p>El usuario deberá proporcionar:</p> <ul style="list-style-type: none">• Un sistema implicacional de entrada sobre el que se aplicará el algoritmo seleccionado. Estos sistemas se podrán seleccionar desde un fichero en disco, o generándolo con el Generador de Implicaciones.• Un fichero de salida en el que se guardará el sistema implicacional resultante devuelto por el algoritmo. Por defecto, el fichero de salida será [workspace]/output/[abreviatura_alg]_output.txt, siendo [workspace] el workspace actual y [abreviatura_alg] el nombre corto del algoritmo.• Uno o varios modos de ejecución que serán:<ul style="list-style-type: none">- Tiempos- Trazas- Estadísticas <p>Una vez ejecutado el algoritmo seleccionado con los parámetros introducidos, se generará un archivo con el sistema implicacional de salida y otros con las trazas según los modos de ejecución seleccionados.</p> <p>Además, se persistirá esta ejecución guardando:</p> <ul style="list-style-type: none">• Fecha / hora de ejecución.• Algoritmo ejecutado.• Tiempo de ejecución (si el modo "Time" fue activado)• Rutas absolutas de los ficheros que contienen: sistema de salida, histórico (si el modo "Trace" ha sido activado) y estadísticas (si el modo "Statistics" ha sido activado).
ID	UC04
Stereotypes	UseCase
Requirements	Diseño de pestaña "Run" en el área Benchmarks., Cargar árbol de benchmarks a partir de los registrados en el workspace

Name	Value
	actual., Inicializar valores por defecto al seleccionar un benchmark, Inicializar valores por defecto al seleccionar un algoritmo, Mostrar cuadro de diálogo para selección de entrada / salida, Ejecutar el algoritmo seleccionado, Ejecutar el benchmark seleccionado, Guardar resultados de la ejecución de un algoritmo.
Preconditions	 Registrar Algoritmo


Ejecutar un algoritmo

1. Acceder a la pantalla "Ejecución de Benchmarks"
2. Seleccionar un algoritmo del árbol de benchmarks.
3. **SYSTEM** Se inicializa el campo "Output" con la ruta [workspace_actual]/output/[abreviatura_algoritmo].txt
4. Introducir el path de un archivo de salida.
5. Seleccionar los modos deseados.
6. Pulsar el botón "Ejecutar".
7. **SYSTEM** El sistema visualiza la traza según el modo de ejecución seleccionado.
8. **SYSTEM** Se guarda en la ruta de salida introducida, el sistema implicacional resultante.
9. **SYSTEM** En el mismo directorio que el archivo de salida, se guardan los ficheros de traza generados.
10. **SYSTEM** Se guarda el resultado de la ejecución para el algoritmo ejecutado en el benchmark al que pertenece el algoritmo.

Ejecutar un algoritmo con un fichero de entrada no existente

1. Seleccionar un algoritmo del árbol de benchmarks.
2. Introducir el path de un fichero no existente como entrada.
3. El sistema muestra un mensaje de error indicando que el fichero de entrada no existe.
4. Corregir el path del fichero de entrada con uno correcto.
5. Pulsar el botón "Ejecutar".
6. **SYSTEM** El sistema visualiza la traza según el modo de ejecución seleccionado.
7. **SYSTEM** Se guarda en la ruta de salida introducida, el sistema implicacional resultante.
8. **SYSTEM** En el mismo directorio que el archivo de salida, se guardan los ficheros de traza generados.
9. **SYSTEM** Se guarda el resultado de la ejecución para el algoritmo ejecutado en el benchmark al que pertenece el algoritmo.


Ejecutar con estadísticas

Name	Value
Description	<p>El usuario seleccionará el modo "Statistics" para la ejecución con traza.</p> <p>En esta ejecución, además de generar un archivo con el sistema implicacional de salida, el sistema genera el archivo [nombre_archivo_salida].csv, en el que se guardan la evolución de los tamaños del sistema implicacional procesado.</p> <p>[nombre_archivo_salida] es el nombre base del archivo seleccionado para la salida del algoritmo. P.e., si el archivo que se ha tomado como salida es <i>do_output.txt</i>, el archivo con los tiempos será <i>do.csv</i>.</p> <p>La información se guarda en archivos .csv para facilitar su visualización mediante tablas y gráficos.</p>
ID	UC17
Stereotypes	UseCase
Preconditions	 Registrar Algoritmo

Ejecutar algoritmo con estadísticas

1. Acceder a la pantalla "Ejecución de Benchmarks"
 2. Seleccionar un algoritmo del árbol de benchmarks.
 3. Introducir el path de un fichero que contenga un sistema implicacional de entrada.
 4. Introducir el path de un archivo de salida.
 5. Seleccionar el modo "Statistics"
 6. Pulsar el botón "Ejecutar".
2. **SYSTEM** Se generar un archivo .log con el tiempo de ejecución y un archivo .csv con los tamaños.


Ejecutar con tiempos

Name	Value
Description	<p>El usuario seleccionará el modo "Time" para la ejecución con tiempos.</p> <p>En esta ejecución, además de generar un archivo con el sistema implicacional de salida, el sistema genera el archivo [nombre_archivo_salida]_history.log, en el que se traza el tiempo de ejecución del algoritmo.</p> <p>[nombre_archivo_salida] es el nombre base del archivo seleccionado para la salida del algoritmo. P.e., si el archivo que se ha tomado como salida es <i>do_output.txt</i>, el archivo con los tiempos será <i>do_history.log</i>.</p>
ID	UC16
Stereotypes	UseCase
Requirements	Guardar el tiempo de ejecución del algoritmo en el resultado de éste. Mostrar el tiempo de ejecución en el visor.
Preconditions	 Registrar Algoritmo

Ejecutar un algoritmo con Tiempos

1. Acceder a la pantalla "Ejecución de Benchmarks"
2. Seleccionar un algoritmo del árbol de benchmarks.
3. **SYSTEM** Se inicializa el campo "Output" con la ruta [workspace_actual]/output/[abreviatura_algoritmo].txt
4. Introducir el path de un archivo con un sistema implicacional de entrada.
5. Seleccionar el modo "Time".
6. Pulsar el botón "Ejecutar".
7. **SYSTEM** Se guarda en la ruta de salida introducida, el sistema implicacional resultante.
8. **SYSTEM** Se guarda el resultado de la ejecución para el algoritmo ejecutado en el benchmark al que pertenece el algoritmo y el tiempo de ejecución.
9. **SYSTEM** Se muestra en el visor el tiempo de ejecución.

Ejecutar con traza

Name	Value
Description	<p>El usuario seleccionará el modo "Trace" para la ejecución con traza.</p> <p>En esta ejecución, además de generar un archivo con el sistema implicacional de salida, el sistema genera el archivo [nombre_archivo_salida]_history.log, en el que se traza el tiempo de ejecución del algoritmo y traza de dicha ejecución. [nombre_archivo_salida] es el nombre base del archivo seleccionado para la salida del algoritmo. P.e., si el archivo que se ha tomado como salida es <i>do_output.txt</i>, el archivo con la traza será <i>do_history.log</i>.</p>
ID	UC15
Stereotypes	UseCase
Requirements	Escribir en un fichero la traza generada por un algoritmo., Mostrar la traza generada en el visor.
Preconditions	 Registrar Algoritmo



Ejecutar un algoritmo con traza y tiempos

1. Acceder a la pantalla "Ejecución de Benchmarks"
2. Seleccionar un algoritmo del árbol de benchmarks.
3. **SYSTEM** Se inicializa el campo "Output" con la ruta [workspace_actual]/output/[abreviatura_algoritmo].txt
4. Introducir el path de un archivo con un sistema implicacional de entrada.
5. Seleccionar los modos "Time" y "Trace"
6. Pulsar el botón "Ejecutar".
7. **SYSTEM** El sistema visualiza la traza según el modo de ejecución seleccionado.
8. **SYSTEM** Se guarda en la ruta de salida introducida, el sistema implicacional resultante.
9. **SYSTEM** Se genera el archivo [nombre_archivo_salida]_history.log en el mismo directorio que el archivo de salida con la traza y tiempo de ejecución..
10. **SYSTEM** Se guarda el resultado de la ejecución para el algoritmo ejecutado en el benchmark al que pertenece el algoritmo.

1.2.4. Resultados



Summary

Name	Description
 Usuario	<p>Usuario general de la aplicación.</p> <p>No existirán roles ni permisos especiales. Cualquier usuario puede realizar cualquier acción.</p>
 Consultar resultados	<p>Para cada ejecución de un algoritmo se guardará un resumen de ésta que contendrá:</p> <ul style="list-style-type: none">• Algoritmo ejecutado.• Fecha y hora.• Sistema de entrada.• Sistema de salida.• Fichero• Tiempo de ejecución. <p>El usuario podrá consultar estos datos a posteriori para comparar las distintas ejecuciones, en una tabla agrupada por benchmarks con n filas, una por algoritmo ejecutado y tres columnas:</p> <ul style="list-style-type: none">- Tiempo de ejecución- Tamaño inicial- Tamaño final

Details



Usuario

Name	Value
Description	Usuario general de la aplicación. No existirán roles ni permisos especiales. Cualquier usuario puede realizar cualquier acción.
ID	AC01
Visibility	public



Consultar resultados

Name	Value
Description	<p>Para cada ejecución de un algoritmo se guardará un resumen de ésta que contendrá:</p> <p>Algoritmo ejecutado.</p> <ul style="list-style-type: none">• Fecha y hora.• Sistema de entrada.• Sistema de salida.• Fichero• Tiempo de ejecución. <p>El usuario podrá consultar estos datos a posteriori para comparar las distintas ejecuciones, en una tabla agrupada por benchmarks con n filas, una por algoritmo ejecutado y tres columnas:</p> <ul style="list-style-type: none">- Tiempo de ejecución- Tamaño inicial- Tamaño final
ID	UC02
Stereotypes	UseCase
Preconditions	Ejecutar benchmark

Consulta de resultados de una ejecución

1. Seleccionar le botón "Results"
2. **SYSTEM** Se muestra una tabla en la que para cada benchmark ejecutado se muestra una fila por algoritmo y para cada uno su tiempo de ejecución, tamaño inicial del sistem y tamaño final.

1.3. Diagrama de componentes

En los diagramas de componentes se muestran los elementos de diseño de un sistema de software. Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces.

En el siguiente diagrama se muestran los principales componentes de la aplicación y las relaciones entre ellos.

- **is-bench:** Herramienta para ejecución de algoritmos y benchmarks sobre sistemas implicacionales.
- **is-algorithms:** API para la implementación y ejecución de algoritmos sobre sistemas implicacionales.
- **implications-generator:** Generador de sistemas implicacionales aleatorios.
- **java-lattice:** Librería de la Dra. K. Bertet, para el manejo de sistemas implicacionales.
- **trace.txt:** Fichero de traza que se genera en la ejecución de un algoritmo.
- **statistics.csv:** Fichero con las estadísticas de los resultados obtenidos en la ejecución de algoritmos.

1.4. Diagramas de paquetes

- **is-bench:** Librerías que implementan la herramienta para ejecución de algoritmos y benchmarks sobre sistemas implicacionales.
 - **is-bench-ui:** Paquete con las clases de la interfaz de usuario.
 - **is-bench-domain:** Paquete con las clases de dominio.
 - **is-bench-business:** Paquete con las clases relacionadas con la persistencia de la información en ficheros.
- **is-algorithms:** API para la implementación y ejecución de algoritmos sobre sistemas implicacionales.
- **implications-generator:** Contiene las clases que implementan el generador de sistemas implicacionales aleatorios.
- **is-commons:** Librería de utilidades genéricas.

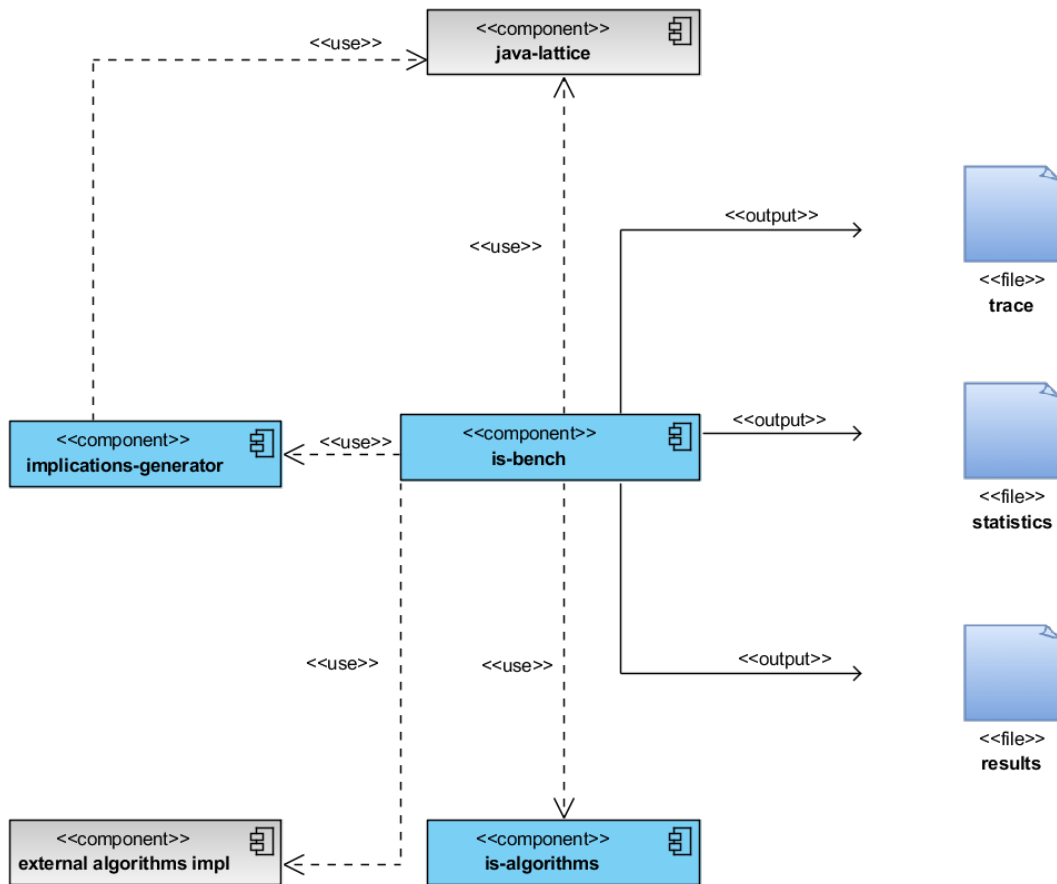


Figura 1.1: Diagrama de componentes

- **java-lattice:** Librería de la Dra. K. Bertet, para el manejo de sistemas impli-
cacionales en la que se basan las librerías anteriores.
- **ch.qos.logback:** Librería de logging, para la generación de traza de la ejecución
de algoritmos y benchmarks (Logback project).
- **guava:** Librería de Google que implementa diversas utilidades (Guava project).
En este proyecto, se utiliza en concreto la implementación del bus de eventos
(Eventbus).

1.5. Diagramas de clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

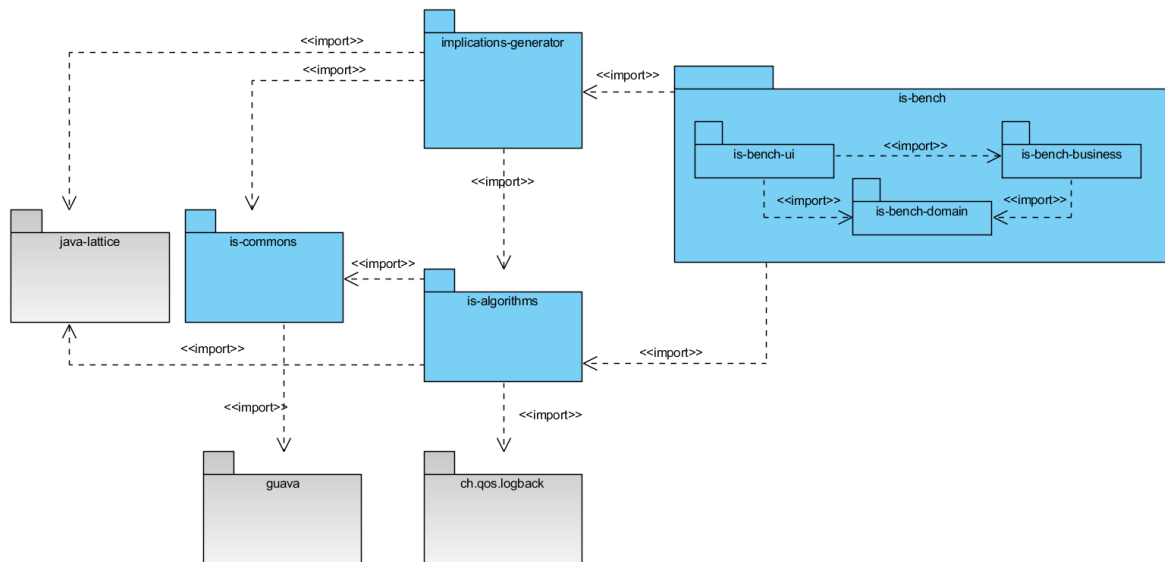


Figura 1.2: Diagrama de paquetes

En esta sección, se presentan los diagramas de clases asociados a cada uno de los casos de uso del apartado anterior y sus especificaciones.

1.5.1. Estereotipos

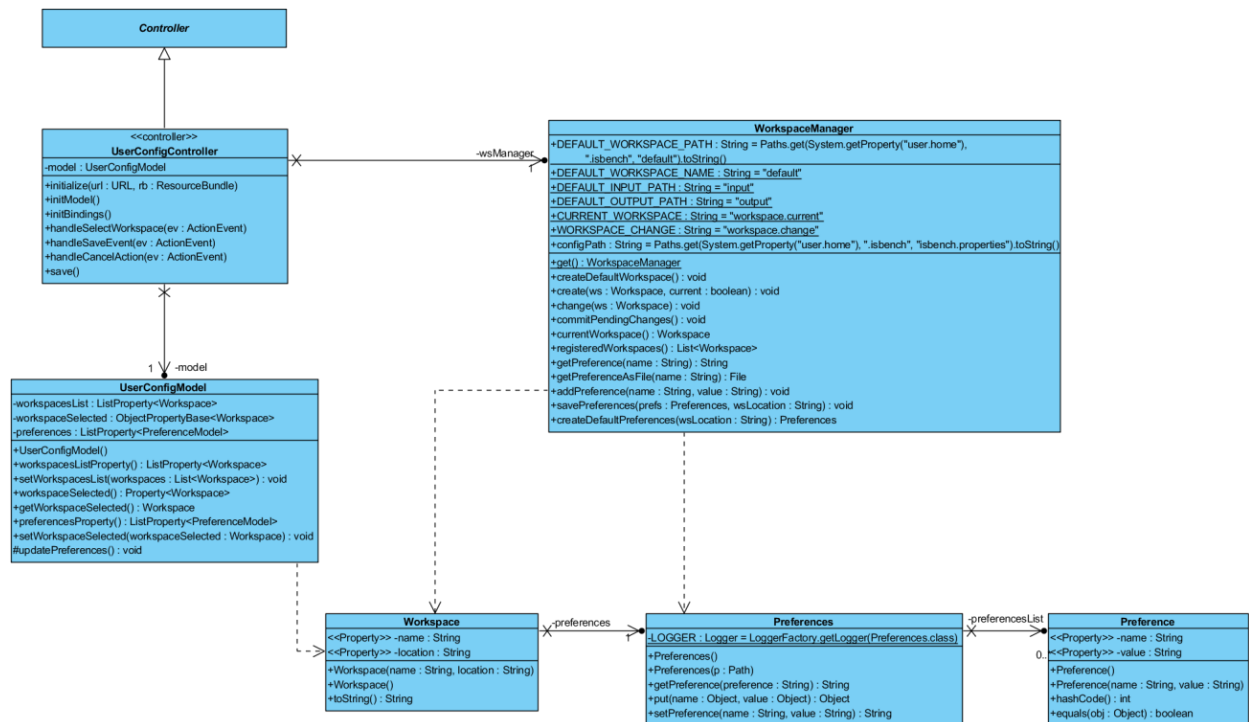
Name	Description
viewpoint	User interface.
controller	Controller of a FXML View.
FXML	FXML component injection.
listener	Method or class which handles component, model or eventbus events.
singleton	Implements Singleton pattern design.

1.5.2. Dominio

\AM@currentdocname .png

.png

1.5.3. Workspaces



Summary

Name	Description
Controller	Controller of a FXML View.
WorkspaceManager	Workspaces manager.
UserConfigController	Workspaces view controller.
UserConfigModel	Workspaces model.
Preference	User preference.
Workspace	Workspace.
Preferences	Contains the user preferences. These are serialized to properties file in the workspace.

Details





WorkspaceManager


Name	Value
Description	Workspaces manager.
Visibility	public


Attributes


<u>private LOGGER : org.slf4j.Logger</u>			
Description	Logger.		
Initial Value	LoggerFactory.getLogger(WorkspaceManager.class)		
Type	org.slf4j.Logger		
Getter	false	Setter	false
Multiplicity	1		


<u>public DEFAULT_WORKSPACE_PATH : String</u>			
Description	Default workspace location.		
Initial Value	Paths.get(System.getProperty("user.home"), ".isbench", "default").toString()		
Type	 String		
Getter	false	Setter	false
Multiplicity	1		

<u>public DEFAULT_WORKSPACE_NAME : String</u>			
Description	Default workspace name.		
Initial Value	"default"		
Type	 String		
Getter	false	Setter	false
Multiplicity	1		


<u>public DEFAULT_INPUT_PATH : String</u>			
Description	Input default directory.		
Initial Value	"input"		
Type	 String		
Getter	false	Setter	false
Multiplicity	1		


<u>public DEFAULT_OUTPUT_PATH : String</u>			
Description	Output default directory.		
Initial Value	"output"		
Type	 String		
Getter	false	Setter	false
Multiplicity	1		


<u>public CURRENT_WORKSPACE : String</u>			
Description	Current workspace property.		
Initial Value	"workspace.current"		
Type	 String		
Getter	false	Setter	false
Multiplicity	1		

<u>public WORKSPACE_CHANGE : String</u>			
Description	Current workspace property.		
Initial Value	"workspace.change"		
Type	 String		
Getter	false	Setter	false
Multiplicity	1		

<u>private config : java.util.Properties</u>			
Description	General configuration.		
Type	java.util.Properties		
Getter	false	Setter	false
Derived	false		
Multiplicity	1		

public configPath : String			
Description	Configuration file path.		
Initial Value	Paths.get(System.getProperty("user.home"), ".isbench", "isbench.properties").toString()		
Type	 String		
Getter	false	Setter	false
Multiplicity	1		

private me : WorkspaceManager			
Description	Single instance.		
Type	 WorkspaceManager		
Getter	false	Setter	false
Multiplicity	1		
Is ID	false		
Leaf	false		

private current : Workspace			
Description	Current workspace.		
Type	 Workspace		
Getter	false	Setter	false
Multiplicity	1		

Operations


protected WorkspaceManager ()	
Description	Constructor. Protected constructor. For gets an instance of WorkspaceManger, will must be use the get() method.

protected WorkspaceManager (configPath : String, config : java.util.Properties)		
Parameters	configPath	
	Description	Configuration file path.
	Multiplicity	1
	Type	● String
	Direction	inout
	config	
	Description	Configuration properties file.
	Multiplicity	1
	Type	java.util.Properties
	Direction	inout
Description	Protected constructor. For gets an instance of WorkspaceManger, will must be use the get() method.	



public get () : WorkspaceManager	
Description	Returns a single instance of WorkspaceManager.
Return Type Description	WorkspaceManager single instance.

protected initialize () : void	
Description	<p>Initializes the workspacemanager.</p> <p>If no exists the config file, creates it.</p> <p>If the current workspace not is setted, creates the default workspace and sets it as the current.</p>

protected initConfig () : void	
Description	Loads the isbench.properties file and if no exists, creates it.
Exceptions	IOException if read / Write error occur.


protected setCurrentWorkspace (ws : Workspace) : void		
Parameters	ws	
	Multiplicity	1
	Type	 Workspace
	Direction	inout
Description	Only for testing purpose.	

public createDefaultWorkspace () : void	
Description	Creates the default workspace.


public create (ws : Workspace, current : boolean) : void		
Parameters	ws	
	Description	Workspace info.
	Multiplicity	1
	Type	 Workspace
	Direction	inout
	current	
	Description	If sets this workspace to the current.
	Multiplicity	1
	Type	 boolean
	Direction	inout
Description	Create the basic workspace directories and files hirerchachy, based on the info that contains the ws param.	


public currentWorkspace () : Workspace	
Description	Returns the current workspace.
Return Type Description	Current workspace.

public registeredWorkspaces () : java.util.List<Workspace>	
Description	Returns the registered workspaces list.
Return Type Description	Workspaces list.

public change (ws : Workspace) : void		
Parameters	ws	
	Description	Workspace.
	Multiplicity	1
	Type	 Workspace
	Direction	inout
Description	Mark a workspace as current. The change will take effect with commitPendingChanges() method.	

public commitPendingChanges () : void	
Description	The workspace change takes effect.

public getPreference (name : String) : String		
Parameters	name	
	Description	Preference name.
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Gets a preference value of the current workspace.	
Return Type Description	Preference value. null if the preference not exists.	

public getPreferenceAsFile (name : String) : java.io.File		
Parameters	name	
	Description	Preference name.
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Assume that a preference value is the file path and returns as it.	
Return Type Description	File given from the consulted preference.	
Exceptions	InvalidPathException if the path string cannot be converted to a Path.	

public addPreference (name : String, value : String) : void		
Parameters	name	
	Description	Name.
	Multiplicity	1
	Type	● String
	Direction	inout
	value	
	Description	Value.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Adds a new preference.	

public savePreferences (prefs : Preferences, wsLocation : String) : void		
Parameters	prefs	
	Description	Preferences.
	Multiplicity	1
	Type	☒ Preferences
	Direction	inout
	wsLocation	
	Description	Directory of the preferences.properties file.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Saves the preferences in a preferences.properties file, in a path (wsLocation param).	
Exceptions	IOException if an error saving the file occur.	

public createDefaultPreferences (wsLocation : String) : Preferences		
Parameters	wsLocation	
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Create a default preferences.	
Return Type Description	Prefrences created.	

protected saveWorkspace (ws : Workspace, current : boolean) : void		
Parameters	ws	
	Description	Workspace.
	Multiplicity	1
	Type	▢ Workspace
	Direction	inout
	current	
	Description	If sets this workspace to the current.
	Multiplicity	1
	Type	● boolean
	Direction	inout
Description	Save workspace in config file.	
Exceptions	IOException if an error saving the file occur.	


protected saveConfig () : void	
Description	Save the config in the file.
Exceptions	IOException if an error saving the file occur.




UserConfigController

Name	Value
Description	Workspaces view controller.
Visibility	public
Stereotypes	controller

Attributes

private model : UserConfigModel			
Description	Workspaces model.		
Type	 UserConfigModel		
Getter	false	Setter	false
Multiplicity	1		

private wsManager : WorkspaceManager			
Description	Workspaces manager.		
Type	 WorkspaceManager		
Getter	false	Setter	false
Multiplicity	1		

Operations

public initialize (url : java.net.URL, rb : java.util.ResourceBundle)		
Parameters	url	
	Description	View URL.
	Multiplicity	1
	Type	java.net.URL
	Direction	inout
	rb	
	Description	Internationalization resources.
	Multiplicity	1
	Type	java.util.ResourceBundle
	Direction	inout
Description	Initializes the controller class.	

public initModel ()	
Description	Initializes the model with the registered workspaces.

public initBindings ()	
Description	Initializes the view-model bindings.

public handleSelectWorkspace (ev : ActionEvent)		
Parameters	ev	
	Description	Action event.
	Multiplicity	1
	Type	ActionEvent
	Direction	inout
Description	Handles the event thrown when <i>Search(...)</i> button is pressed. Open a dialog for directory selection. The selection shows in the Location field.	
Stereotypes	FXML	

public handleSaveEvent (ev : ActionEvent)		
Parameters	ev	
	Description	Action event.
	Multiplicity	1
	Type	ActionEvent
	Direction	inout
Description	Handles the event thrown when Save button is pressed. Invokes the save() method to save the changes and closes	
Stereotypes	FXML	

public handleCancelAction (ev : ActionEvent)		
Parameters	ev	
	Description	Action event.
	Multiplicity	1
	Type	ActionEvent
	Direction	inout
Description	Ignores the changes and closes the window.	
Stereotypes	FXML	

public save ()	
Description	Saves the workspace change. If the workspace is new, asks to the user if he would like establish it as the current workspace. If the user accepts, shows a new message indication that the change will take effect in the next application start.



UserConfigModel

Name	Value
Description	UserConfig view model.
Visibility	public

Attributes

private workspacesList : javafx.beans.property.ListProperty			
Description	Workspaces list.		
Type	javafx.beans.property.ListProperty		
Getter	false	Setter	false
Multiplicity	1		

private workspaceSelected : javafx.beans.property.ObjectPropertyBase			
Description	Workspace selected binding property.		
Type	javafx.beans.property.ObjectPropertyBase		
Getter	false	Setter	false
Multiplicity	1		

private preferences : javafx.beans.property.ListProperty			
Description	Preferences of selected workspace binding property.		
Type	javafx.beans.property.ListProperty		
Getter	false	Setter	false
Multiplicity	1		

Operations

public UserConfigModel ()	
Description	Constructor.

public workspacesListProperty () : javafx.beans.property.ListProperty	
Description	Workspaces list binding property.
Return Type Description	Workspaces list binding property.


public setWorkspacesList (workspaces : java.util.List<Workspace>) : void		
Parameters	workspaces	
	Description	Workspaces paths.
	Multiplicity	0..*
	Type	java.util.List<Workspace>
	Direction	inout
Description	Sets the workspaces list.	

public workspaceSelected () : javafx.beans.property.Property	
Description	Selected workspace binding property.
Return Type Description	Selected workspace binding property.

public getWorkspaceSelected () : Workspace	
Description	Gets the selected workspace from workspaceSelected property.
Return Type Description	The selected workspace.

public preferencesProperty () : javafx.beans.property.ListProperty	
Description	Preferences list binding property.
Return Type Description	Preferences list binding property.

protected updatePreferences () : void	
Description	Updates the preferences with the selected workspace's preferences.


public setWorkspaceSelected (workspaceSelected : Workspace) : void		
Parameters	workspaceSelected	
	Description	The workspaceSelected to set.
	Multiplicity	1
	Type	 Workspace
	Direction	inout
Description	Sets the selected workspace.	




Preference

Name	Value
Description	User preference.
Visibility	public

Attributes

private name : String			
Description	Name.		
Stereotypes	Property		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		

private value : String			
Description	Value.		
Stereotypes	Property		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		

Operations

public Preference ()	
Description	Constructor.

public Preference (name : String, value : String)		
	name	
	Description	Name.
	Multiplicity	1
	Type	● String
	Direction	inout
	value	
	Description	Value.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Constructor.	


public hashCode () : int	
Description	Compute the hash code based on the name property.
Return Type Description	Hash code.


public equals (obj : Object) : boolean		
Parameters	obj	
	Description	Object to compare.
	Multiplicity	1
	Type	Object
	Direction	inout
Description	Two Preference instances are equal if their names are.	
Return Type Description	True if the obj parameter is a Preference instance, and its names is equal to this instance name.	


Workspace

Name	Value
Description	Workspace.
Visibility	public

Attributes

private name : String			
Description	Name.		
Stereotypes	Property		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		

private location : String			
Description	Location.		
Stereotypes	Property		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		

private preferences : Preferences			
Description	User preferences in the workspace.		
Stereotypes	Property		
Type	 Preferences		
Getter	true	Setter	true
Multiplicity	1		

Operations

public Workspace (name : String, location : String)		
Parameters	name	
	Multiplicity	1
	Type	● String
	Direction	inout
	location	
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Creates a workspace with the name and location specified.	

public Workspace ()	
Description	Constructor.


public toString () : String	
Description	Workspace string representation, this is the absolute path of its location.
Return Type Description	Workspace string representation.



Preferences

Name	Value
Description	Contains the user preferences. These are serialized to properties file in the workspace.
Visibility	public

Attributes

private preferencesList : Preference			
Description	Preferences list.		
Stereotypes	Property		
Type	 Preference		
Getter	true	Setter	false
Multiplicity	0..*		

private LOGGER : org.slf4j.Logger			
Description	Logger.		
Initial Value	LoggerFactory.getLogger(Preferences.class)		
Type	org.slf4j.Logger		
Getter	false	Setter	false
Derived	false		
Multiplicity	1		

Operations

public Preferences ()	
Description	Constructor.

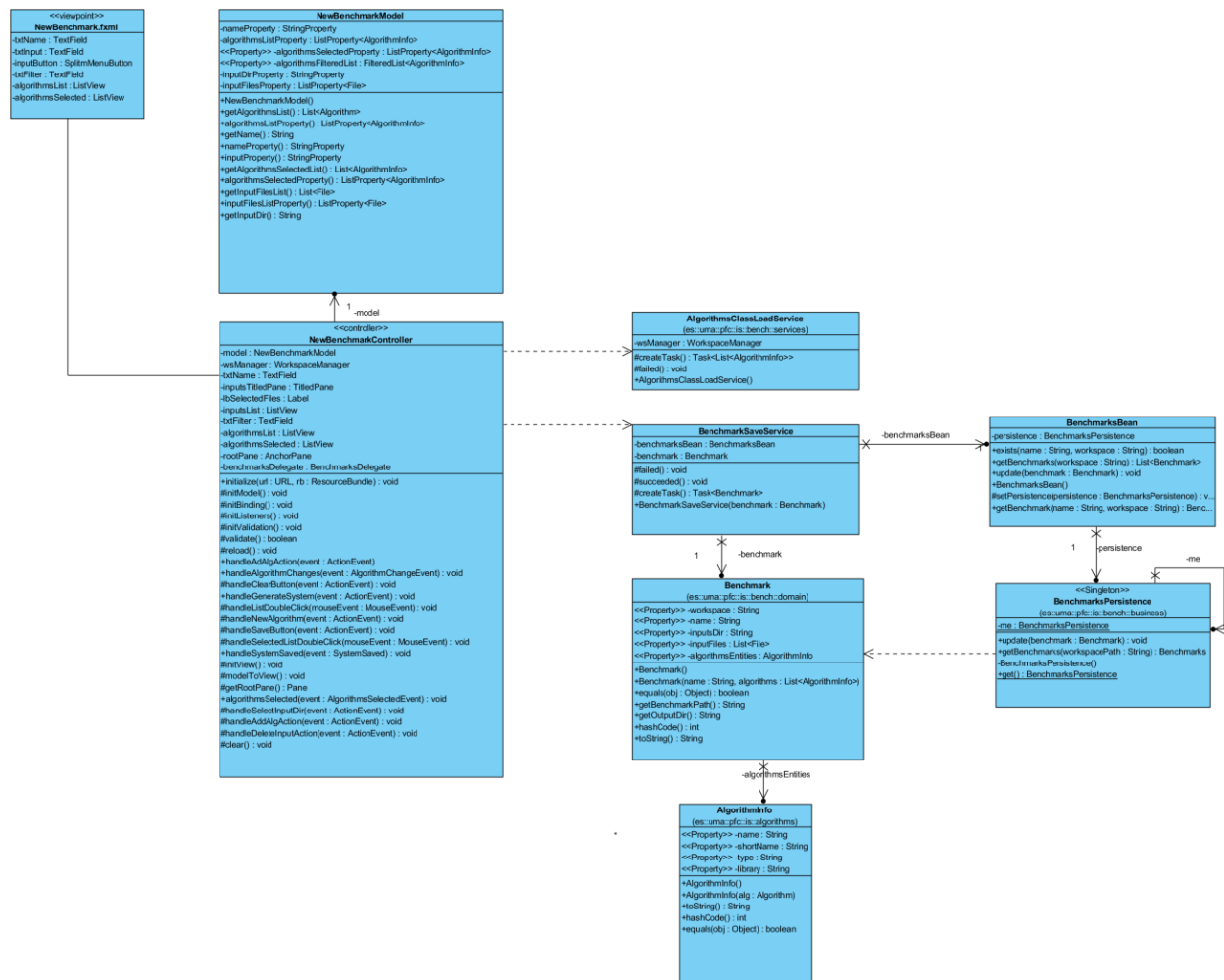
public Preferences (p : java.nio.file.Path)		
Parameters	p	
	Description	Preferences file absolute path.
	Multiplicity	1
	Type	java.nio.file.Path
	Direction	inout
Description	Builds an instance based on an existent preferences file.	

public getPreference (preference : String) : String		
Parameters	preference	
	Description	Preference name.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Preference value.	
Return Type Description	Preference value.	










public put (name : Object, value : Object) : Object		
Parameters	name	
	Description	Name.
	Multiplicity	1
	Type	Object
	Direction	inout
	value	
	Description	Value.
	Multiplicity	1
	Type	Object
	Direction	inout
Description	Creates or modify a preference.	
Return Type Description	Previous value of modified preference.	

public setPreference (name : String, value : String) : String		
Parameters	name	
	Multiplicity	1
	Type	● String
	Direction	inout
	value	
	Description	Value.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Sets the value of a preference. If value is null, remove the preference if exists.	

1.5.4. Registrar Benchmarks



Summary

Name	Description
 NewBenchmark.fxml	New benchmark view.
 NewBenchmarkModel	New Benchmark view model.
 AlgorithmsClassLoadService	Loads the algorithms found in the libraries included in the workspace lib folder.
 NewBenchmarkController	NewBenchmark view Controller class.
 BenchmarksBean	Business logic for insert, modify and delete algorithms.
 BenchmarkSaveService	Service which saves benchmark changes.
 Benchmark	Benchmark entity.
 BenchmarksPersistence	Persist the benchmarks into an XML file entities using JAXB.
 AlgorithmInfo	Entity with an algorithm attributes.

Description

Class diagram that shows the relation between the classes involved in the registry of benchmarks.

Details



NewBenchmark.fxml

Name	Value
Description	New benchmark view.
Visibility	public
Stereotypes	viewpoint

Attributes

private txtName : javafx.scene.control.TextField			
Description	Name field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private txtInput : javafx.scene.control.TextField			
Description	Absolute path of input implicational system file.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private inputButton : javafx.scene.control.SplitMenuButton			
Description	Button for select the input implicational system file. It has two options: select the file or generate a random system.		
Type	javafx.scene.control.SplitMenuButton		
Getter	false	Setter	false
Multiplicity	1		

private txtFilter : javafx.scene.control.TextField			
Description	Search algorithms field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private algorithmsList : javafx.scene.control.ListView			
Description	Registered algorithms list in the current workspace.		
Type	javafx.scene.control.ListView		
Getter	false	Setter	false
Multiplicity	1		

private algorithmsSelected : javafx.scene.control.ListView			
Description	List which contains the selected algorithms from registered algorithms list.		
Type	javafx.scene.control.ListView		
Getter	false	Setter	false
Multiplicity	1		



NewBenchmarkController

Name	Value
Description	NewBenchmark view controller.
Visibility	public
Stereotypes	controller

Attributes

private model : NewBenchmarkModel			
Description	Model.		
Type	NewBenchmarkModel		
Getter	false	Setter	false
Multiplicity	1		

private txtName : javafx.scene.control.javafx.scene.control.TextField			
Description	Name field.		
Type	javafx.scene.control.javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private inputsTitledPane : javafx.scene.control.TitledPane			
Description	Container of inputs implicational systems selected.		
Type	javafx.scene.control.TitledPane		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private lbSelectedFiles : javafx.scene.control.Label			
Description	Label with selected files count.		
Type	javafx.scene.control.Label		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		


private inputsList : javafx.scene.control.ListView			
Description	List of selected input files.		
Type	javafx.scene.control.ListView		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		


private txtFilter : javafx.scene.control.textfield.TextField			
Description	Algorithms filter.		
Type	javafx.scene.control.textfield.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private algorithmsList : javafx.scene.control.ListView			
Description	Available algorithms.		
Type	javafx.scene.control.ListView		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private algorithmsSelected : javafx.scene.control.ListView			
Description	Algorithms selected.		
Type	javafx.scene.control.ListView		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private rootPane : javafx.scene.layout.AnchorPane			
Description	Root pane.		
Type	javafx.scene.layout.AnchorPane		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private wsManager: WorkspaceManager			
Description	Workspace manager.		
Type	 WorkspaceManager		
Getter	false	Setter	false
Multiplicity	1		

private benchmarksDelegate : BenchmarksDelegate			
Description	Delegate for benchmarks business logic.		
Type	 BenchmarksDelegate		
Getter	false	Setter	false
Multiplicity	1		

Operations

public initialize (url : java.net.URL, rb : java.util.ResourceBundle) : void			
Parameters	url		
	Description	URL of the view.	
	Multiplicity	1	
	Type	java.net.URL	
	Direction	inout	
	rb		
	Description	Resource bundle.	
	Multiplicity	1	
	Type	java.util.ResourceBundle	
	Direction	inout	
	Description	Initializes the controller class.	

protected initModel () : void	
Description	Initializes the model.

protected initBinding () : void	
Description	Initializes the binding between view components and the model.

protected initListeners () : void	
Description	Initializes the components view listeners.


protected initValidation () : void	
Description	Initializes the validation support.

protected initView () : void	
Description	Initializes the inputsList selection mode.
Exceptions	IOException

protected modelToView () : void	
Description	Load the algorithms list with algorihtms of the model.

protected validate () : boolean	
Description	Form validations.
Return Type Description	<code>true</code> if there is one algorithm selected at least, and the benchmark name not exists or user wants override it, and parent validations is succeeded. <code>False</code> otherwise.

protected reload () : void	
Description	Reloads the view and model.

public handleAlgorithmChanges (event : AlgorithmChangeEvent) : void		
Parameters	event	
	Description	Event thrown when an algorithms is created or modified.
	Multiplicity	1
	Type	 AlgorithmChangeEvent
	Direction	inout
Description	Handles the AlgorithmChangeEvent published by the Eventbus. Reloads the model and view.	

protected handleClearButton (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Action event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	When the <i>Clear</i> button is pressed, the fields are cleared.	
Stereotypes	FXML	


public handleGenerateSystem (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event thrown when the <i>Random</i> option of <i>Input</i> button is pressed.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Shows the generator panel for generate a random system.	
Stereotypes	FXML	

protected handleListDoubleClick (mouseEvent : javafx.scene.input.MouseEvent) : void		
Parameters	mouseEvent	
	Description	Mouse event.
	Multiplicity	1
	Type	javafx.scene.input.MouseEvent
	Direction	inout
Description	When there is a double click in algorithms list, the selection is added to algorithms selected.	
Stereotypes	FXML	


protected handleNewAlgorithm (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Action event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Handles the event thrown when <i>New Algorithm</i> button is pressed.	
Stereotypes	FXML	

protected handleSaveButton (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Action event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	When the <i>Save</i> button is pressed, the current values are validated and saved.	
Stereotypes	FXML	

protected handleSelectedListDoubleClick (mouseEvent : javafx.scene.input.MouseEvent) : void		
Parameters	mouseEvent	
	Description	Mouse event.
	Multiplicity	1
	Type	javafx.scene.input.MouseEvent
	Direction	inout
Description	When there is a double click in selected algorithms list, the selection is remove from algorithms selected.	
Stereotypes	FXML	

public handleSystemSaved (event : SystemSaved) : void		
Parameters	event	
	Description	EventBus event.
	Multiplicity	1
	Type	 SystemSaved
	Direction	inout
Description	Handles the <i>SystemSaved</i> event, copying the path of system into input field.	

protected getRootPane () : javafx.scene.layout.Pane	
Description	Root pane.
Return Type Description	Pane.

public algorithmsSelected (event : AlgorithmsSelectedEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	 AlgorithmsSelectedEvent
	Direction	inout
Description	Handles the AlgorithmsSelectedEvent published by the Eventbus. Adds all algorithms contained in the event to algorithms selected list.	

protected handleSelectInputDir (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event thrown when the File option of Select Input button is pressed.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Shows a file chooser for select the input system file.	
Stereotypes	FXML	

protected handleAddAlgAction (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event thrown when the <i>Add Algorithm</i> button is pressed.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	When the <i>Add Algorithm</i> button is pressed, the <i>New Algorithm</i> window is shown.	
Stereotypes	FXML	

protected clear () : void	
Description	Clears all fields.

protected handleDeleteInputAction (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event thrown when the contextual menu <i>Delete</i> option is pressed.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Deletes the current selection in the inputs list.	





NewBenchmarkModel

Name	Value
Description	New Benchmark view model.
Visibility	public

Attributes


private nameProperty : javafx.beans.property.StringProperty			
Description	Name binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private algorithmsListProperty : javafx.beans.property.ListProperty			
Description	Algorithms list binding property.		
Type	 javafx.beans.property.ListProperty		
Getter	false	Setter	false
Multiplicity	1		

private algorithmsSelectedProperty : javafx.beans.property.ListProperty			
Description	Selected algorithms list binding property.		
Type	 javafx.beans.property.ListProperty		
Getter	false	Setter	true
Multiplicity	1		

private algorithmsFilteredList : javafx.collections.transformation.FilteredList			
Description	Algorithms filtered list.		
Stereotypes	Property		
Type	javafx.collections.transformation.FilteredList		
Getter	true	Setter	false
Multiplicity	1		

private inputDirProperty : javafx.beans.property.StringProperty			
Description	Input directory path binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private inputFilesProperty : javafx.beans.property.ListProperty			
Description	Input files list binding property.		
Type	 javafx.beans.property.ListProperty		
Getter	false	Setter	false
Multiplicity	1		

Operations

public NewBenchmarkModel ()	
Description	Constructor.

public nameProperty () : javafx.beans.property.StringProperty	
Description	Name.
Return Type Description	The name binding property.

public getName () : String	
Description	Gets the benchmark name from the nameProperty.
Return Type Description	Benchmark name.

public inputProperty () : javafx.beans.property.StringProperty	
Description	Input files directory binding property.
Return Type Description	Input files directory binding list property.

public getInputDir () : java.lang.String	
Description	Gets the input files directory path from the inputDirProperty binding property.
Return Type Description	Input files directory absolute path.

public inputFilesListProperty () : javafx.beans.property.ListProperty	
Description	Input files list binding property.
Return Type Description	Input files binding list property.

public getInputFilesList () : java.util.List	
Description	Gets the input files list from the inputFilesListProperty binding property.
Return Type Description	Algorithms list.

public algorithmsListProperty () : javafx.beans.property.ListProperty	
Description	Algorithms list binding property.
Return Type Description	Algorithms list binding property.

public getAlgorithmsList () : java.util.List	
Description	Gets the algorithms list from the algorithmsListProperty binding property.
Return Type Description	Current workspace registered algorithms.

public algorithmsSelectedProperty () : javafx.beans.property.ListProperty	
Description	Algorithms list binding property.
Return Type Description	The algorithmsSelectedProperty.

public getAlgorithmsSelectedList () : java.util.List	
Description	Gets the selected algorithms list from the algorithmsSelectedProperty binding property.
Return Type Description	Selected algorithms.



AlgorithmsClassLoadService

Name	Value
Description	Loads the algorithms found in the libraries included in the workspace lib folder.
Visibility	public

Attributes

private wsManager : WorkspaceManager			
Description	Workspace manager.		
Type	WorkspaceManager		
Getter	false	Setter	false
Multiplicity	1		

Operations

public AlgorithmsLoadService ()	
Description	Constructor.

protected createTask () : javafx.concurrent.Task	
Description	Creates the background task which loads the found algorithms.
Return Type Description	Algorithms info list.


protected failed () : void	
Description	This method is executed when the background task is completed with errors.




BenchmarkSaveService


Name	Value
Description	Service which saves benchmark changes.
Visibility	public

Attributes

private benchmarksBean : BenchmarksBean			
Description	Benchmarks logic.		
Type	 BenchmarksBean		
Getter	false	Setter	false
Multiplicity	1		

private benchmark : Benchmark			
Description	Benchmark to save.		
Type	 Benchmark		
Getter	false	Setter	false
Multiplicity	1		

Operations

public BenchmarkSaveService (benchmark : Benchmark)		
Parameters	benchmark	
	Description	Benchmark to save.
	Multiplicity	1
	Type	 Benchmark
	Direction	inout
Description	Constructor.	

protected createTask () : javafx.concurrent.Task	
Description	Creates the background task which saves the benchmarks changes.
Return Type Description	Updated benchmark.


protected succeeded () : void	
Description	This method is executed when the background task is completed succesfully. Publishes a BenchmarksChangeEvent and MessageEvent by the Eventbus.

protected failed () : void	
Description	This method is executed when the background task is completed with errors.

BenchmarksBean


Name	Value
Description	Business logic for insert, modify and delete algorithms.
Visibility	public


Attributes


private persistence : BenchmarksPersistence			
Description	Class for benchmarks persistence.		
Type	 BenchmarksPersistence		
Getter	false	Setter	false
Multiplicity	1		



Operations

public BenchmarksBean ()	
Description	Constructor.

protected setPersistence (persistence : BenchmarksPersistence) : void		
Parameters	persistence	
	Multiplicity	1
	Type	 BenchmarksPersistence
	Direction	inout
Description	For testing usage.	

public update (benchmark : Benchmark) : void		
Parameters	benchmark	
	Description	Benchmark.
	Multiplicity	1
	Type	 Benchmark
	Direction	inout
Description	Create the directory tree of benchmark.	
Exceptions	java.io.IOException	

public getBenchmarks (workspace : String) : java.util.List		
Parameters	workspace	
	Description	Workspace path.
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Returns the workspace registered benchmarks.	
Return Type Description	Registered benchmarks.	
Exceptions	java.lang.Exception	

public getBenchmark (name : String, workspace : String) : Benchmark		
Parameters	name	
	Description	Benchmark name.
	Multiplicity	1
	Type	 String
	Direction	inout
	workspace	
	Description	Workspace path.
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Returns a benchmark registered in a workspace, null if no exists.	


public getBenchmark (name : String, workspace : String) : Benchmark	
Return Type Description	Algorithm Entities list.
Exceptions	java.lang.Exception

public exists (name : String, workspace : String) : boolean		
Parameters	name	
	Description	Benchmark name.
	Multiplicity	1
	Type	● String
	Direction	inout
	workspace	
	Description	Workspace path.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	If exists a benchmark with the name argument in a workspace.	
Return Type Description	true if exists a benchmark with the name argument, false otherwise.	

BenchmarksPersistence

Name	Value
Description	Persist the benchmarks into an XML file entities using JAXB.
Visibility	public
Stereotypes	Singleton


Attributes


private me : BenchmarksPersistence			
Description	Single instance.		
Type	 BenchmarksPersistence		
Getter	false	Setter	false
Multiplicity	1		


Operations


private BenchmarksPersistence ()	
Description	Private constructor. For get a BenchmarksPersistence instance, will must be usage the static get() method.

public get () : BenchmarksPersistence	
Description	Gets a single instance of BenchmarksPersistence.
Return Type Description	BenchmarksPersistence single instance.

public getBenchmarks (workspacePath : String) : Benchmarks		
Parameters	workspacePath	
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Returns the registered benchmarks in a workspace.	

public update (benchmark : Benchmark) : void		
Parameters	benchmark	
	Description	Algorithms.
	Multiplicity	1
	Type	 Benchmark
	Direction	inout
Description	Initialize the benchmarks file with benchmarks parameter.	


public insert (algorithms : Algorithms) : void		
Parameters	algorithms	
	Multiplicity	1
	Type	 Algorithms
	Direction	inout
Description	Add the algorithms of algorithms parameter to algorithms file.	


public insert (algorithm : AlgorithmInfo) : void		
Parameters	algorithm	
	Description	Algorithm.
	Multiplicity	1
	Type	 AlgorithmInfo
	Direction	inout
Description	Add an algorithm to algorithms file.	


Benchmark

Name	Value
Description	Benchmark entity.
Visibility	public
Stereotypes	XmlRootElement


Attributes

private workspace : String			
Description	Workspace which the benchmark is registered.		
Stereotypes	Property, XmlAttribute		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		

private name : String			
Description	Benchmark Name.		
Stereotypes	Property, XmlAttribute		
Type	 String		
Getter	true	Setter	false
Multiplicity	1		

private inputsDir : String			
Description	Input implicational systems dir path.		
Stereotypes	Property, XmlAttribute		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		

private inputFiles : java.util.List			
Description	Input files which will be copied to input dir.		
Stereotypes	Property		
Type	java.util.List, XmlTransient		
Getter	true	Setter	true
Multiplicity	1..*		

private algorithmsEntities : java.util.List<AlgorithmInfo>			
Description	Benchmark algorithms.		
Stereotypes	Property, XmlElement		
Type	 AlgorithmInfo		
Getter	true	Setter	false
Multiplicity	*		

Operations

public Benchmark ()	
Description	Constructor.

public Benchmark (name : String, algorithms : java.util.List)	
Description	Constructor.
Exceptions	IllegalArgumentException if the name or algorithms list are empty.

public getBenchmarkPath () : String	
Description	The benchmark path.
Return Type Description	Benchmark path.

public getOutputDir () : String	
Description	Path of output directory of benchmark.
Return Type Description	Output directory path.

public equals (obj : Object) : boolean		
Parameters	obj	
	Multiplicity	1
	Type	Object
	Direction	inout
Description	Two benchmarks are equals if their names are.	
Return Type Description	true if the obj parameter is a benchmarks and its name is equal to name of this benchmark instance, false otherwise.	


public hashCode () : int	
Description	Compute the hashCode based on the name property.
Return type description	HashCode.


public toString () : String	
Description	Benchmark string representation.
Return type description	Benchmark's name.


AlgorithmInfo

Name	Value
Description	Entity with an algorithm attributes.
Visibility	public
Stereotypes	XmlRootElement

Attributes


private name : String			
Description	Name.		
Stereotypes	Property		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		
Java Detail	N/A		

private shortName : String			
Description	Short name.		
Stereotypes	Property		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		

private type : Class			
Description	Algorithm implementation class.		
Stereotypes	Property		
Type	 java.lang.Class		
Getter	true	Setter	true
Multiplicity	1		

Operations

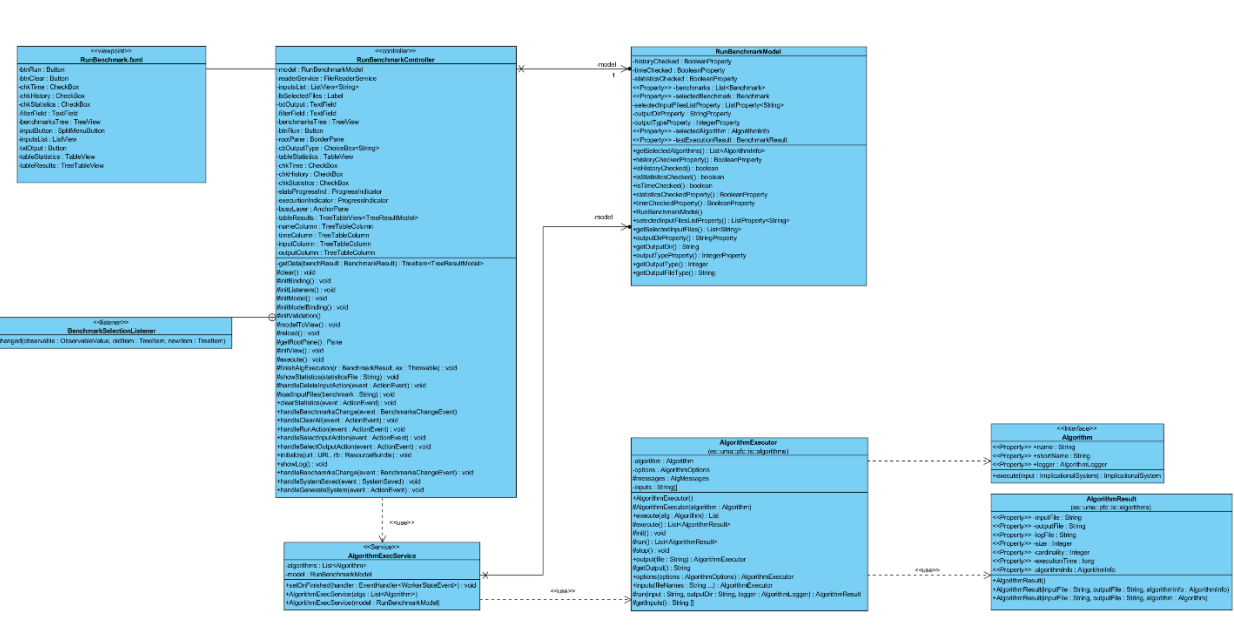
public AlgorithmInfo ()	
Description	Constructor.
Upper	1








public AlgorithmInfo (alg : Algorithm)		
Parameters	alg	
	Description	Algorithm.
	Multiplicity	1
	Type	 Algorithm
	Direction	inout
Description	Constructs an AlgorithmInfo instance from an algorithm implementation.	
Upper	1	

public toString () : String	
Description	
Return type description	

public hashCode () : int	
Description	
Return type description	

public equals (obj : Object) : boolean		
Parameters	obj	
	Multiplicity	Unspecified
	Type	Object
	Direction	inout
Description	Two AlgorithmInfo objets are equals if their names an short names are.	
Return type description	true if the obj parameter is a AlgorithmInfo instance and its name and short name are equal to name and short name of this benchmark instance, false otherwise.	



Name	Description
 RunBenchmark.fxml	Benchmarks execution view.
 RunBenchmarkController	Run view controller.
 BenchmarkSelectionListener	Benchmarks tree listener.
 RunBenchmarkModel	Benchmarks execution model.
 AlgorithmExecService	Service for the background execution of algorithms and benchmarks.
 AlgorithmExecutor	Service which executes an algorithm.
 Algorithm	Algorithm of implicational system basis computation. It can be found in <i>API Algoritmos</i> class diagram.

 RunBenchmark.fxml

Name	Value
Description	Benchmarks execution view.
Visibility	public
Stereotypes	viewpoint

Attributes

private btnRun : javafx.scene.control.Button			
Type	javafx.scene.control.Button		
Getter	false	Setter	false
Multiplicity	1		

private btnClear : javafx.scene.control.Button			
Type	javafx.scene.control.Button		
Getter	false	Setter	false
Multiplicity	1		

private chkTime : javafx.scene.control.CheckBox			
Description	Check for to enable / disable the Time mode.		
Type	javafx.scene.control.CheckBox		
Getter	false	Setter	false
Multiplicity	1		

private chkHistory : javafx.scene.control.CheckBox			
Description	Check for to enable / disable the Trace mode.		
Type	javafx.scene.control.CheckBox		
Getter	false	Setter	false
Multiplicity	1		

private chkStatistics : javafx.scene.control.CheckBox			
Description	Check for to enable / disable the Statistics mode.		
Type	javafx.scene.control.CheckBox		
Getter	false	Setter	false
Multiplicity	1		

private filterField : javafx.scene.control.TextField			
Description	Filter over the benchmarks tree.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private benchmarksTree : javafx.scene.control.TreeView			
Description	Two levels tree with the current workspace registered benchmarks at the first level, and its algorithms at the second level.		
Type	javafx.scene.control.TreeView		
Getter	false	Setter	false
Multiplicity	1		

private inputButton : javafx.scene.control.SplitMenuButton			
Description	Menu button which shows the dialog box for to select the input files. Contains two options: select the existent file or generate a random implicational system.		
Type	javafx.scene.control.SplitMenuButton		
Getter	false	Setter	false
Multiplicity	1		

private inputsList : javafx.scene.control.ListView			
Description	Input implicational system file paths list.		
Type	javafx.scene.control.ListView		
Getter	false	Setter	false
Multiplicity	1		
Leaf	false		

private txtOutput : javafx.scene.control.Button			
Description	Text field for the output directory generated by the benchmark execution.		
Type	javafx.scene.control.Button		
Getter	false	Setter	false
Multiplicity	1		

private tableStatistics : javafx.scene.control.TableView			
Description	Table which contains the sizes statistics if the Statistics mode is enabled.		
Type	javafx.scene.control.TableView		
Getter	false	Setter	false
Multiplicity	1		

private tableResults : javafx.scene.control.TreeTableView			
Description	Table which contains the results of the executed benchmark, grouped by benchmark and algorithm.		
Type	javafx.scene.control.TreeTableView		
Getter	false	Setter	false
Multiplicity	1		



RunBenchmarkController

Name	Value
Description	RunBenchmark view controller.
Visibility	public

Attributes

private model : RunBenchmarkModel			
Description	RunBenchmark view model.		
Type	RunBenchmarkModel		
Getter	false	Setter	false
Multiplicity	1		

private readerService : FileReaderService			
Description	Service which reads a file in background.		
Type	FileReaderService		
Getter	false	Setter	false
Multiplicity	1		

private inputsList : javafx.scene.control.ListView			
Description	Input implicational system file paths list.		
Type	javafx.scene.control.ListView		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private lbSelectedFiles : javafx.scene.control.Label			
Description	Input selected files selected count.		
Type	javafx.scene.control.Label		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private txtOutput : javafx.scene.control.TextField			
Description	Output field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private filterField : javafx.scene.control.TextField			
Description	Filter over benchmarks tree.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private benchmarksTree : javafx.scene.control.javafx.scene.control.TreeView			
Description	Benchmarks and algorithms tree.		
Type	javafx.scene.control.javafx.scene.control.TreeView		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private btnRun : javafx.scene.control.Button			
Description	Run button.		
Type	javafx.scene.control.Button		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private rootPane : javafx.scene.layout.BorderPane			
Description	Root pane.		
Type	javafx.scene.layout.BorderPane		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private cbOutputType : javafx.scene.control.ChoiceBox			
Description	Output type dropdown.		
Type	javafx.scene.control.ChoiceBox		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private tableStatistics : javafx.scene.control.TableView			
Description	Table with statistics results.		
Type	javafx.scene.control.TableView		
Stereotypes	FXML		
Getter	false	Setter	false
Multiplicity	1		

private chkTime : javafx.scene.control.CheckBox			
Description	Time mode check.		
Type	javafx.scene.control.CheckBox		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private chkHistory : javafx.scene.control.CheckBox			
Description	Trace mode check.		
Type	javafx.scene.control.CheckBox		
Getter	false	Setter	false
Multiplicity	Unspecified		
Stereotypes	FXML		

private chkStatistics : javafx.scene.control.CheckBox			
Description	Statistics mode check.		
Type	javafx.scene.control.CheckBox		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private statsProgressInd : javafx.scene.control.ProgressIndicator			
Description	Statistics loading progress indicator.		
Type	javafx.scene.control.ProgressIndicator		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private executionIndicator : javafx.scene.control.ProgressIndicator			
Description	Execution progress indicator.		
Type	javafx.scene.control.ProgressIndicator		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private busyLayer : javafx.scene.layout.AnchorPane			
Description	Layer which contains progress indicators.		
Type	javafx.scene.layout.AnchorPane		
Getter	false	Setter	false
Multiplicity	1		

private tableResults : javafx.scene.control.TreeTableView			
Description	Execution results table.		
Type	javafx.scene.control.TreeTableView		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private nameColumn : javafx.scene.control.TreeTableColumn			
Description	Columns of execution results table.		
Type	javafx.scene.control.TreeTableColumn		
Getter	false	Setter	false
Multiplicity	1		

private timeColumn : javafx.scene.control.TreeTableColumn			
Description	Columns of execution results table.		
Type	javafx.scene.control.TreeTableColumn		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private inputColumn : javafx.scene.control.TreeTableColumn			
Description	Input file column of execution results table.		
Type	javafx.scene.control.TreeTableColumn		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private outputColumn : javafx.scene.control.TreeTableColumn			
Description	Columns of execution results table.		
Type	javafx.scene.control.TreeTableColumn		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

Operations

protected initModel () : void	
Description	Initializes the model loading the benchmarks into the tree.

protected initView () : void	
Description	Initializes the view.
Exceptions	IOException


protected initBinding () : void	
Description	Initializes the bindings between the components view and the model.

protected initListeners () : void	
Description	Initializes the components view listeners.

protected initModelBinding () : void	
Description	Initializes the bindings with the model.

public initialize (url : java.net.URL, rb : java.util.ResourceBundle) : void		
Parameters	url	
	Description	URL of the view.
	Multiplicity	1
	Type	java.net.URL
	Direction	inout
	rb	
	Description	Resource bundle.
	Multiplicity	1
	Type	java.util.ResourceBundle
	Direction	inout
Description	Initializes the controller.	

protected modelToView () : void	
Description	The components that are not bound , will be updated with the model values.

private getData (benchResult : BenchmarkResult) : javafx.scene.control.javafx.scene.control.TreeItem		
Parameters	benchResult	
	Description	Benchmark results.
	Multiplicity	1
	Type	 BenchmarkResult
	Direction	inout
Description	Loads the benchmark results into a TreeTableView hierarchy.	
Return Type Description	Benchmark result node.	

protected reload () : void	
Description	Reloads the view and model.


protected getRootPane () : javafx.scene.layout.Pane	
Description	Returns the root pane.
Return Type Description	Root pane.

protected clear () : void	
Description	Clears the model and the view.

public clearStatistics (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Clears the statistics table.	

public showLog () : void	
Description	Shows the algorithm result selected log.


protected execute () : void	
Description	Executes the algorithms or benchmark selected with AlgorithmExecService.

public handleBenchmarksChange (event : BenchmarksChangeEvent) : void		
Parameters	event	
	Description	Registered benchmarks change event.
	Multiplicity	1
	Type	 BenchmarksChangeEvent
	Direction	inout
Description	Handles the event BenchmarksChangeEvent published by Eventbus, and reloads the view and the model.	
Stereotypes	FXML	

public handleClearAll (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	ActionEvent handler of Run button. Clear all traces.	
Stereotypes	FXML	


public handleRunAction (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Handles the event thrown when Run button is pressed. Runs the selected benchmark / algorithm.	
Stereotypes	FXML	


public handleSelectInputAction (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Shows a dialog box for select the input of algorithm.	

public handleSystemSaved (event : SystemSaved) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	 SystemSaved
	Direction	inout
Description	Handles the SystemSaved event, copying the path of system into input field.	


public handleGenerateSystem (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Action event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Shows the generator window for generate a random system.	
Stereotypes	FXML	


public handleSelectOutputAction (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Shows the dialog box for select the target of algorithm results.	
Stereotypes	FXML	

public handleBenchamrksChange (event : BenchmarksChangeEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	 BenchmarksChangeEvent
	Direction	inout
Description	Handles the event BenchmarksChangeEvent published by Eventbus, and reloads view and model.	

protected loadInputFiles (benchmark : String) : void		
Parameters	benchmark	
	Description	Benchmark's name.
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Loads the input files defined for the selected benchmark.	

protected handleDeleteInputAction (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Removes the selected items from the list.	
Stereotypes	FXML	

protected finishAlgExecution (result : BenchmarkResult, ex : Throwable) : void		
Parameters	result	
	Description	Benchmark result.
	Multiplicity	1
	Type	 BenchmarkResult
	Direction	inout
	ex	
	Description	Exception if the execution is failed. Null otherwise.
	Multiplicity	1
	Type	Throwable
	Direction	inout
Description	Shows the results and statistics.	

protected showStatistics (statisticsFile : String) : void		
Parameters	statisticsFile	
	Description	Statistics file path.
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Loads the result statistics into a table.	



RunBenchmarkModel

Name	Value
Description	Benchmarks execution model.
Visibility	public


Attributes

private historyChecked : javafx.beans.property.BooleanProperty			
Description	Trace mode checked binding property.		
Type	javafx.beans.property.BooleanProperty		
Getter	false	Setter	false
Multiplicity	1		

private timeChecked : javafx.beans.property.BooleanProperty			
Description	Time mode checked binding property.		
Type	javafx.beans.property.BooleanProperty		
Getter	false	Setter	false
Multiplicity	1		

private statisticsChecked : javafx.beans.property.BooleanProperty			
Description	Statistics mode checked binding property.		
Type	javafx.beans.property.BooleanProperty		
Getter	false	Setter	false
Multiplicity	1		


private benchmarks : java.util.List<Benchmark>			
Description	Available benchmarks.		
Stereotypes	Property		
Type	java.util.List<Benchmark>		
Getter	true	Setter	true
Multiplicity	0..*		


private selectedBenchmark : Benchmark			
Description	Selected benchmark.		
Stereotypes	Property		
Type	 Benchmark		
Getter	true	Setter	true
Multiplicity	1		

private selectedInputFilesListProperty : javafx.beans.property.ListProperty			
Description	Input files list binding property.		
Type	javafx.beans.property.ListProperty		
Getter	false	Setter	false
Multiplicity	1		

private outputDirProperty : javafx.beans.property.StringProperty			
Description	Output file path binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private outputTypeProperty : javafx.beans.property.IntegerProperty			
Description	Output file type selected binding property.		
Type	javafx.beans.property.IntegerProperty		
Getter	false	Setter	false
Multiplicity	1		

private selectedAlgorithm : AlgorithmInfo			
Description	When the selection is an algorithm, the selected algorithm info.		
Stereotypes	Property		
Type	 AlgorithmInfo		
Getter	true	Setter	true
Multiplicity	1		

private lastExecutionResult : BenchmarkResult			
Description	Last execution result.		
Stereotypes	Property		
Type	 BenchmarkResult		
Getter	true	Setter	true
Multiplicity	1		

Operations

public RunBenchmarkModel ()	
Description	Constructor.

public getSelectedAlgorithms () : java.util.List<AlgorithmInfo>	
Description	Selected algorithms.
Return Type Description	List of selected algorithms info.

public historyCheckedProperty () : javafx.beans.property.BooleanProperty	
Description	Trace checked binding property.
Return Type Description	The trace checked binding property.

public isHistoryChecked () : boolean	
Description	If the trace mode is checked.
Return Type Description	true if the history mode is checked, false otherwise.

public isStatisticsChecked () : boolean	
Description	If the statistics mode is checked.
Return Type Description	true if the statistics mode is checked, false otherwise.

public isTimeChecked () : boolean	
Description	If the time mode is checked.
Return Type Description	true if the Time mode is checked, false otherwise.

public statisticsCheckedProperty () : javafx.beans.property.BooleanProperty	
Description	Statistics checked binding property.
Return Type Description	The statistics checked binding property.

public timeCheckedProperty () : javafx.beans.property.BooleanProperty	
Description	Time checked binding property.
Return Type Description	The time checked binding property.

public selectedInputFilesListProperty () : javafx.beans.property.ListProperty	
Description	Path input files binding property.
Return Type Description	The Path input files binding property.

public getSelectedInputFiles () : java.util.List<String>	
Description	Gets the selected input files paths from the selectedInputFilesListProperty.
Return Type Description	The selected input files paths.

public outputDirProperty () : javafx.beans.property.StringProperty	
Description	Path output directory binding property.
Return Type Description	The Path output directory property.

public getOutputDir () : String	
Description	Returns the output directory path.
Return Type Description	The output directory path.

public outputTypeProperty () : javafx.beans.property.IntegerProperty	
Description	Output type binding property.
Return Type Description	The Output type binding property.

public getOutputType () : Integer	
Description	Gets the output file type from the outputTypeProperty.
Return Type Description	The output type.

public getOutputFileType () : String	
Description	Returns the file type of the selected output type.
Return Type Description	File Type.



RunBenchmarkController.BenchmarkSelectionListener

Name	Value
Description	Benchmarks tree listener.
Visibility	protected
Stereotypes	listener

Operations


public changed (observable : ObservableValue, oldItem : javafx.scene.control.TreeItem, newItem : javafx.scene.control.TreeItem)		
Parameters	observable	
	Multiplicity	1
	Type	ObservableValue
	Direction	inout
	oldItem	
	Multiplicity	1
	Type	javafx.scene.control.TreeItem
	Direction	inout
	newItem	
	Multiplicity	1
	Type	javafx.scene.control.TreeItem
	Direction	inout
Description	<p>This method is invoked when the selection on the benchmarks tree is changed.</p> <ul style="list-style-type: none"> • When a benchmark is selected, the output field is initialized with the benchmark output directory and is disabled. • In this case, the mode checks and console outputs are disabled too. • If the selection is an algorithm, the output field is enabled and initialized with the path of algorithm default output file. • If the selection is more than one algorithm, the output field is cleared and disabled. • If the selection contains benchmarks and algorithms the output field is cleared and the Run button disabled. 	

AlgorithmExecService

Name	Value
Description	Service for the background execution of algorithms and benchmarks.
Visibility	public

Attributes

private algorithms : java.util.List<Algorithm>			
Description	Algorithms to execute.		
Type	java.util.List<Algorithm>		
Getter	false	Setter	false
Multiplicity	1..*		

private model : RunBenchmarkModel			
Description	Model.		
Type	 RunBenchmarkModel		
Getter	false	Setter	false
Multiplicity	1		

Operations

protected createTask () : javafx.concurrent.Task	
Description	<p>Task for execute algorithms and benchmarks.</p> <p>If the Statistics mode is enabled, prints the results into a CSV file.</p>

protected instanceAlgorithms () : void	
Description	Creates the instances of the benchmark algorithms.


protected getOptions () : AlgorithmOptions	
Description	Establishes the algorithm options from the model.
Return Type Description	Algorithm execution options.


public setOnFinished (handler : javafx.event.EventHandler) : void		
Parameters	handler	
	Description	Handler.
	Multiplicity	1
	Type	javafx.event.EventHandler
	Direction	inout
Description	The onFinish event handler is called whenever the Task state transitions to the finished state: CANCELLED, FAILED or SUCCEEDED.	


protected failed () : void	
Description	This method is executed when the background task is completed with errors.

protected succeeded () : void	
Description	Publishes a message event when the task is finished successfully.

public AlgorithmExecService (algs : java.util.List<Algorithm>)		
Parameters	algs	
	Description	Algorithms to execute.
	Multiplicity	1
	Type	java.util.List<Algorithm>
	Direction	inout
Description	Constructor.	

public AlgorithmExecService (model : RunBenchmarkModel)		
Parameters	model	
	Description	Model.
	Multiplicity	1
	Type	 RunBenchmarkModel
	Direction	inout
Description	Constructor.	


protected instanceAlgorithm (algorithm : AlgorithmInfo) : Algorithm		
Parameters	algorithm	
	Description	Algorithm info.
	Multiplicity	1
	Type	 AlgorithmInfo
	Direction	inout
Description	Instances the algorithm of type of algorithm parameter.	
Return Type Description	Algorithm instance.	


protected printResults (result : BenchmarkResult) : void		
Parameters	result	
	Description	Results.
	Multiplicity	1
	Type	 BenchmarkResult
	Direction	inout
Description	Prints the results into a CSV file.	
Exceptions	IOException when a read /write error occur.	


AlgorithmExecutor


Name	Value
Description	Service which executes an algorithm.
Visibility	public

Attributes

private algorithm : Algorithm			
Description	Algorithm to execute.		
Type	 Algorithm		
Getter	false	Setter	false
Multiplicity	1		


private options : AlgorithmOptions			
Description	Execution options.		
Type	 AlgorithmOptions		
Getter	false	Setter	false
Multiplicity	1		


protected messages : AlgMessages			
Description	I18n messages.		
Type	 AlgMessages		
Getter	false	Setter	false
Multiplicity	1		

private inputs : String			
Description	Paths of inputs system.		
Type	 String		
Getter	false	Setter	false
Multiplicity	0..*		

Operations

public AlgorithmExecutor ()	
Description	Constructor.

protected AlgorithmExecutor (algorithm : Algorithm)		
Parameters	algorithm	
	Description	Algorithm to execute.
	Multiplicity	Unspecified
	Type	 Algorithm
	Direction	inout
Description	Protected constructor. For testing purpose only.	

public execute (alg : Algorithm) : java.util.List		
Parameters	alg	
	Description	Algorithm.
	Multiplicity	1
	Type	 Algorithm
	Direction	inout
Description	Executes an algorithm.	
Return Type Description	Algorithm results.	


protected execute () : java.util.List<AlgorithmResult>	
Description	Executes an algorithm in three stages: initialization, execution and finalization.
Return Type Description	Execution results.


protected init () : void	
Description	Initializes the algorithm execution.

protected run () : java.util.List<AlgorithmResult>	
Description	Executes the algorithm with the inputs and options stablished.
Return Type Description	Algorithm results.
Exceptions	<ul style="list-style-type: none"> • java.io.IOException if IO error occur. • java.lang.IllegalArgumentException if the algorithm is null.




public output (file : String) : AlgorithmExecutor		
Parameters	file	
	Description	Path of the output.
	Multiplicity	Unspecified
	Type	String
	Direction	inout
Description	Sets the output path of the result execution.	
Return Type Description	AlgorithmExecutor with output system established.	

protected getOutput () : String	
Description	Returns the output path.
Return Type Description	Output path.

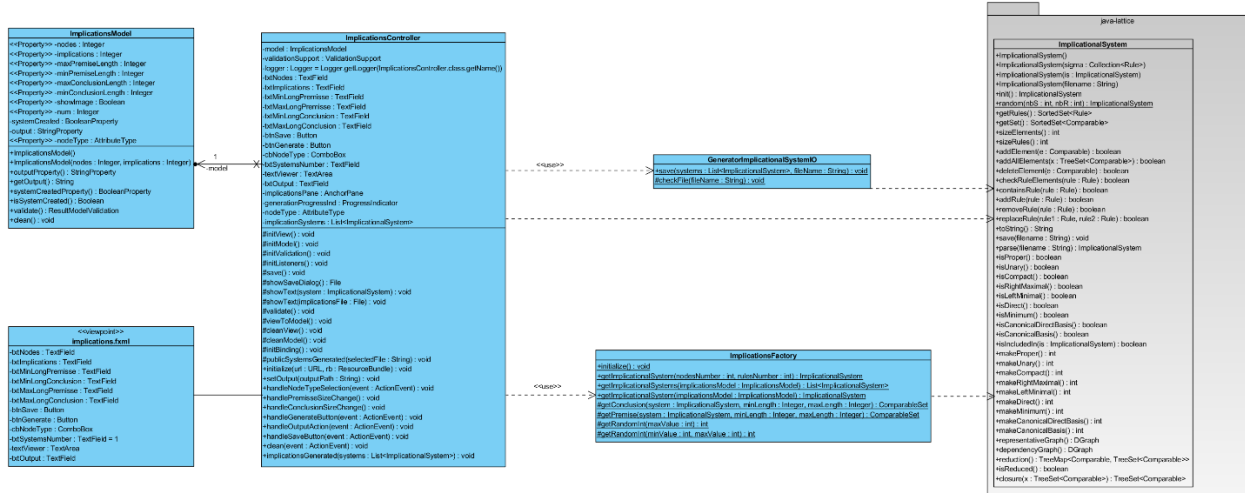
public options (options : AlgorithmOptions) : AlgorithmExecutor		
Parameters	options	
	Description	Execution options.
	Multiplicity	Unspecified
	Type	 AlgorithmOptions
	Direction	inout
Description	Sets an execution options.	
Return Type Description	AlgorithmExecutor with an execution option established.	

public inputs (fileNames : String) : AlgorithmExecutor		
Parameters	fileNames	
	Description	Additional inputs.
	Multiplicity	0..*
	Type Modifier	...
	Type	 String
	Direction	inout
Description	Sets the path of the inputs system.	
Return Type Description	AlgorithmExecutor with inputs system setted.	







protected getInputs () : String	
Description	Returns the inputs path.
Return Type Description	Inputs path.

protected run (input : String, outputDir : String, logger : AlgorithmLogger) : AlgorithmResult		
	input	
	Description	Input implicational system.
	Multiplicity	Unspecified
	Type	 String
	Direction	inout
	outputDir	
	Description	Output dir.
	Multiplicity	Unspecified
	Type	 String
	Direction	inout
	logger	
	Description	Logger.
	Multiplicity	Unspecified
	Type	 AlgorithmLogger
	Direction	inout
Description	Executes the algorithm with an input and output dir.	
Return Type Description	Algorithm result.	

1.5.6. Generador Implicaciones



Summary

Name	Description
 implications.fxml	Implications generator main view.
 ImplicationsController	Implications generator view's controller.
 ImplicationsModel	Class which represents the features of implicational system to generate.
 fr.kbertet.lattice.ImplicationalSystem	This class gives a representation for an implicational system (fr.kbertet.lattice.ImplicationalSystem), a set of rules. It belongs to the java-lattice library.
 ImplicationsFactory	Implicational systems factory.
 GeneratorImplicationalSystemIO	Implements the methods to save implicational systems in files.

Details



implications.fxml

Name	Value
Description	Implications generator main view.
Visibility	public
Stereotypes	viewpoint

Attributes

private txtNodes : TextField			
Description	System nodes number field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private txtImplications : javafx.scene.control.TextField			
Description	Implications number field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private txtMinLongPremisse : javafx.scene.control.TextField			
Description	Premises min length field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private txtMinLongConclusion : javafx.scene.control.TextField			
Description	Conclusions min length field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private txtMaxLongPremisse : javafx.scene.control.TextField			
Description	Premises max length field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private txtMaxLongConclusion : javafx.scene.control.TextField			
Description	Conclusions max length field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private btnSave : javafx.scene.control.Button			
Description	Save button.		
Type	javafx.scene.control.Button		
Getter	false	Setter	false
Multiplicity	1		

private btnGenerate : javafx.scene.control.Button			
Description	Generate button.		
Type	javafx.scene.control.Button		
Getter	false	Setter	false
Multiplicity	1		

private cbNodeType : javafx.scene.control.ComboBox			
Description	Dropdown with the node types: numerics (0, 1,, 2, ...), alphabeticals (a, b, c, ...) or alphanumerics (a0, a1, a2, ...).		
Type	javafx.scene.control.ComboBox		
Getter	false	Setter	false
Multiplicity	1		

private txtSystemsNumber : javafx.scene.control.TextField			
Description	Systems number field.		
Initial Value	1		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private textViewer : javafx.scene.control.TextArea			
Description	Viewer.		
Type	javafx.scene.controlTextArea		
Getter	false	Setter	false
Multiplicity	1		


private txtOutput : javafx.scene.control.TextField			
Description	Path output file field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		



ImplicationsController

Name	Value
Description	Implications generator view's controller.
Visibility	public

Attributes

private model : ImplicationsModel			
Description	Model.		
Type	 ImplicationsModel		
Getter	false	Setter	false
Multiplicity	1		

private validationSupport : org.controlsfx.validation.ValidationSupport			
Description	Validation support.		
Type	org.controlsfx.validation.ValidationSupport		
Getter	false	Setter	false
Multiplicity	1		

private logger : java.util.logging.Logger			
Initial Value	Logger.getLogger(ImplicationsController.class.getName())		
Type	java.util.logging.Logger		
Getter	false	Setter	false
Multiplicity	1		

private txtNodes : javafx.scene.control.TextField			
Description	Nodes number field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private txtImplications : javafx.scene.control.TextField			
Description	Implications number field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private txtMinLongPremisse : javafx.scene.control.TextField			
Description	Min length premise field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private txtMaxLongPremisse : javafx.scene.control.TextField			
Description	Max length premise field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		

private txtMinLongConclusion : javafx.scene.control.TextField			
Description	Min length conclusion field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private txtMaxLongConclusion : javafx.scene.control.TextField			
Description	Max length conclusion field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private btnSave : javafx.scene.control.javafx.scene.control.Button			
Description	Save button.		
Type	javafx.scene.control.javafx.scene.control.Button		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private btnGenerate : javafx.scene.control.javafx.scene.control.Button			
Description	Generate button.		
Type	javafx.scene.control.javafx.scene.control.Button		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private cbNodeType : javafx.scene.control.ComboBox			
Description	Node types dropdown.		
Type	javafx.scene.control.ComboBox		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		


private txtSystemsNumber : javafx.scene.control.TextField			
Description	Number systems to generate field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private textViewer : javafx.scene.control.TextArea			
Description	Generated systems viewer.		
Type	javafx.scene.control.TextArea		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private txtOutput : javafx.scene.control.TextField			
Description	Output file path field.		
Type	javafx.scene.control.TextField		
Getter	false	Setter	false
Stereotypes	FXML		

private implicationsPane : javafx.scene.layout.AnchorPane			
Description	Root pane.		
Type	javafx.scene.layout.AnchorPane		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private generationProgressInd : javafx.scene.control.ProgressIndicator			
Description	System generation progress indicator.		
Type	javafx.scene.control.ProgressIndicator		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private nodeType : AttributeType			
Description	Selected node type.		
Type	 AttributeType		
Getter	false	Setter	false
Multiplicity	1		

private implicationSystems : java.util.List<fr.kbertet.lattice.ImplicationalSystem>			
Description	Generated system.		
Type	java.util.List<fr.kbertet.lattice.ImplicationalSystem>		
Getter	false	Setter	false
Derived	false		
Multiplicity	0..*		

Operations

public initialize (url : java.net.URL, rb : java.util.ResourceBundle) : void			
Parameters	url		
	Description	URL of the view.	
	Multiplicity	1	
	Type	java.net.URL	
	Direction	inout	
	rb		
	Description	Resource Bundle.	
	Multiplicity	1	
	Type	java.util.ResourceBundle	
	Direction	inout	
	Description	Initializes the controller class.	

protected initView () : void		
Description	Initializes the view.	

protected initModel () : void		
Description	Creates an instance of ImplicationsModel.	

protected viewToModel () : void		
Description	Reads the view values and saves them into the model.	

protected initBindingjav() : void		
Description	Initializes the bindings.	

protected initValidation () : void		
Description	Initializes the validation support.	

protected initListeners () : void		
Description	Initializes the listeners.	

public setOutput (outputPath : String) : void		
Parameters	outputPath	
	Description	Output file path.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Sets the output file path into the output field.	

public handlePremisseSizeChange () : void		
Description	Cleans the max and min premise length fields decorators. This method is invoked by the max and min premise fields listeners.	

public handleConclusionSizeChange () : void	
Description	Cleans the max and min conclusion length fields' decorators. This method is invoked by the max and min conclusion fields listeners.

public handleNodeTypeSelection (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	When a node type is selected from the node types dropdown, the node type field is updated.	
Stereotypes	FXML	

public handleOutputAction (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event thrown when de <i>Search (...)</i> button is pressed.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	When the <i>Search (...)</i> button is pressed, opens the dialog box to select an output file, and copy the selected file path into Output field.	
Stereotypes	FXML	

public clean (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event thrown when <i>Clean</i> button is pressed.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Handles the event thrown when the Clean button is pressed and cleans the form fields.	
Stereotypes	FXML	

public handleGenerateButton (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Handles the event thrown when the <i>Generate</i> button is pressed and generates the implications with the inserted values.	
Stereotypes	FXML	

public handleSaveButton (event : javafx.event.ActionEvent) : void		
Parameters	event	
	Description	Even thrown when the <i>Save</i> button is pressed.
	Multiplicity	1
	Type	javafx.event.ActionEvent
	Direction	inout
Description	Saves the implicational system in the path inserted into Output filed.	
Stereotypes	FXML	


protected cleanView () : void	
Description	Cleans the view fields.

protected cleanModel () : void	
Description	Cleans the model values.

protected validate () : void	
Description	Performs the model validation.
Exceptions	RuntimeException If validation error exists.

protected save () : void	
Description	Saves the generated implicational systems with the GeneratorImplicationalSystemIO class.

protected showSaveDialog () : java.io.File	
Description	Shows the <i>Save</i> dialog box and returns the selected file.
Return Type Description	Selected file.

protected showText (system : fr.kbertet.lattice.ImplicationalSystem) : void		
Parameters	system	
	Description	Implicational system.
	Multiplicity	1
	Type	 fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
Description	Shows in the viewer an implicational system.	

protected showText (implicationsFile : java.io.File) : void		
Parameters	implicationsFile	
	Description	Implicational system.
	Multiplicity	1
	Type	java.io.File
	Direction	inout
Description	Shows in the viewer a saved system.	

public implicationsGenerated (systems : java.util.List<Implications>) : void		
Parameters	systems	
	Description	Generated implications.
	Multiplicity	1
	Type	java.util.List<Implications>
	Direction	inout
Description	Shows the generated system in the viewer, if was generated only one. If was generated several, shows the <i>Save</i> dialog box.	

protected publicSystemsGenerated (selectedFile : String) : void		
Parameters	selectedFile	
	Description	Output file path.
	Multiplicity	1
	Type	String
	Direction	inout
Description	Publishes a SystemSaved event by Eventbus.	

ImplicationsModel

Name	Value
Description	Class which represents the features of implicational system to generate.
Visibility	public

Attributes

private nodes : Integer			
Description	Nodes number.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	true
Multiplicity	1		

private implications : Integer			
Description	Implications number.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	true
Multiplicity	1		

private maxPremiseLength : Integer			
Description	Premise attributes max length.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	true
Multiplicity	1		

private minPremiseLength : Integer			
Description	Premise attributes min length.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	true
Multiplicity	1		

private maxConclusionLength : Integer			
Description	Conclusion attributes max length.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	true
Multiplicity	1		


private minConclusionLength : Integer			
Description	Conclusion attributes min length.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	true
Multiplicity	1		

private showImage : Boolean			
Description	If the graph will be painted.		
Stereotypes	Property		
Type	Boolean		
Getter	true	Setter	true
Multiplicity	1		

private num : Integer			
Description	Number of systems to generate.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	true
Multiplicity	1		

private systemCreated : javafx.beans.property.BooleanProperty			
Description	System created flag binding property.		
Type	javafx.beans.property.BooleanProperty		
Getter	false	Setter	false
Multiplicity	1		

private output : javafx.beans.property.StringProperty			
Description	Output file path binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private nodeType : AttributeType			
Description	Node types. It can be numeric, alphabetical or alphanumeric.		
Stereotypes	Property		
Type	 AttributeType		
Getter	true	Setter	true
Multiplicity	1		

Operations

public ImplicationsModel ()	
Description	Constructor.

public ImplicationsModel (nodes : Integer, implications : Integer)		
	nodes	
	Description	Nodes number
	Multiplicity	1
	Type	Integer
	Direction	inout
	implications	
	Description	Implications number.
	Multiplicity	1
	Type	Integer
	Direction	inout
	Description	Constructor.

public outputProperty () : javafx.beans.property.StringProperty	
Description	Returns the output file path binding property.
Return Type Description	The output file path binding property.

public getOutput () : String	
Description	Gets the output file path from the outputProperty property.
Return Type Description	Ouptut file path.

public systemCreatedProperty () : javafx.beans.property.BooleanProperty	
Description	Returns the system created flag binding property.
Return Type Description	The system created flag binding property.

public isSystemCreated () : Boolean	
Description	Gets if the system has been created from the systemCreatedProperty property.
Return Type Description	true if the system has been created, false otherwise.

public validate () : ResultModelValidation	
Description	Checks if the model is correct for the implicational system generation.
Return Type Description	Object with the validation results.

public clean () : void	
Description	Cleans the properties values.






ImplicationsFactory


Name	Value
Description	Random implicational systems factory.
Visibility	public



Operations


<u>public initialize () : void</u>	
Description	Initializes the factory.


<u>public getImplicationalSystems (implicationsModel : ImplicationsModel) : java.util.List</u>		
Parameters	implicationsModel	
	Description	Restrictions of the systems to generate.
	Multiplicity	1
	Type	 ImplicationsModel
	Direction	inout
Description	Returns n implicational systems fulfilling the established restrictions in the implicationsModel parameter.	
Return Type Description	Implicational systems list.	


<u>public getImplicationalSystem (nodesNumber : int, rulesNumber : int) : fr.kbertet.lattice.ImplicationalSystem</u>		
Parameters	nodesNumber	
	Description	Nodes number.
	Multiplicity	1
	Type	 int
	Direction	inout
	rulesNumber	
	Description	Implications number.
	Multiplicity	1
	Type	 int
	Direction	inout
Description	Generates a random system with the nodes and implications number passed by parameter.	
Return Type Description	Generated implicational system.	

<u>public getImplicationalSystem (implicationsModel : ImplicationsModel) : fr.kbertet.lattice.ImplicationalSystem</u>		
Parameters	implicationsModel	
	Description	Restrictions of the systems to generate.
	Multiplicity	1
	Type	 ImplicationsModel
	Direction	inout
Description	Returns an implicational systems fulfilling the established restrictions in the implicationsModel parameter.	
Return Type Description	Random implicational system.	

<u>protected getRandomInt (minValue : int, maxValue : int) : int</u>		
Parameters	minValue	
	Description	Min value.
	Multiplicity	1
	Type	 int
	Direction	inout
	maxValue	
	Description	Max value.
	Multiplicity	1
	Type	 int
	Direction	inout
Description	Returns a random integer, between a min and max value passed as parameters.	
Return Type Description	Random integer value.	

<u>protected getRandomInt (maxValue : int) : int</u>		
Parameters	maxValue	
	Description	Max value.
	Multiplicity	1
	Type	 int
	Direction	inout
Description	Returns a random integer, between 0 and max Value parameter value.	

<u>protected getConclusion (system : fr.kbertet.lattice.ImplicationalSystem, minLength : Integer, maxLength : Integer) : ComparableSet</u>		
Parameters	system	
	Description	Implicational system.
	Multiplicity	1
	Type	 fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	minLength	
	Description	Min nodes number.
	Multiplicity	1
	Type	Integer
	Direction	inout
	maxLength	
	Description	Max nodes number.
	Multiplicity	1
	Type	Integer
	Direction	inout
Description	Returns a conclusion with a max nodes number. If max Length parameter is null or less than 0, the max nodes number is the system nodes number.	
Return Type Description	Conclusion with a max nodes number.	
Exceptions	RuntimeException if the max Length parameter is greater than attributes number.	

<u>protected getPremise (system : fr.kbertet.lattice.ImplicationalSystem, minLength : Integer, maxLength : Integer) : ComparableSet</u>	
Parameters	system
	Description Implicational system.
	Multiplicity 1
	Type  fr.kbertet.lattice.Implicational System
	Direction inout
	minLength
	Description Min nodes number.
	Multiplicity 1
	Type Integer
	Direction inout
	maxLength
	Description Max nodes number.
	Multiplicity 1
	Type Integer
	Direction inout
Description	Returns a premise with a max nodes number. If the maxLength parameter is null or less than 0, the max nodes number is the system nodes number.
Return Type Description	Premise with a max nodes number.
Return Type Description	Random integer value.

GeneratorImplicationalSystemIO

Name	Value
Description	Implements the methods to save implicational systems in files.
Visibility	public

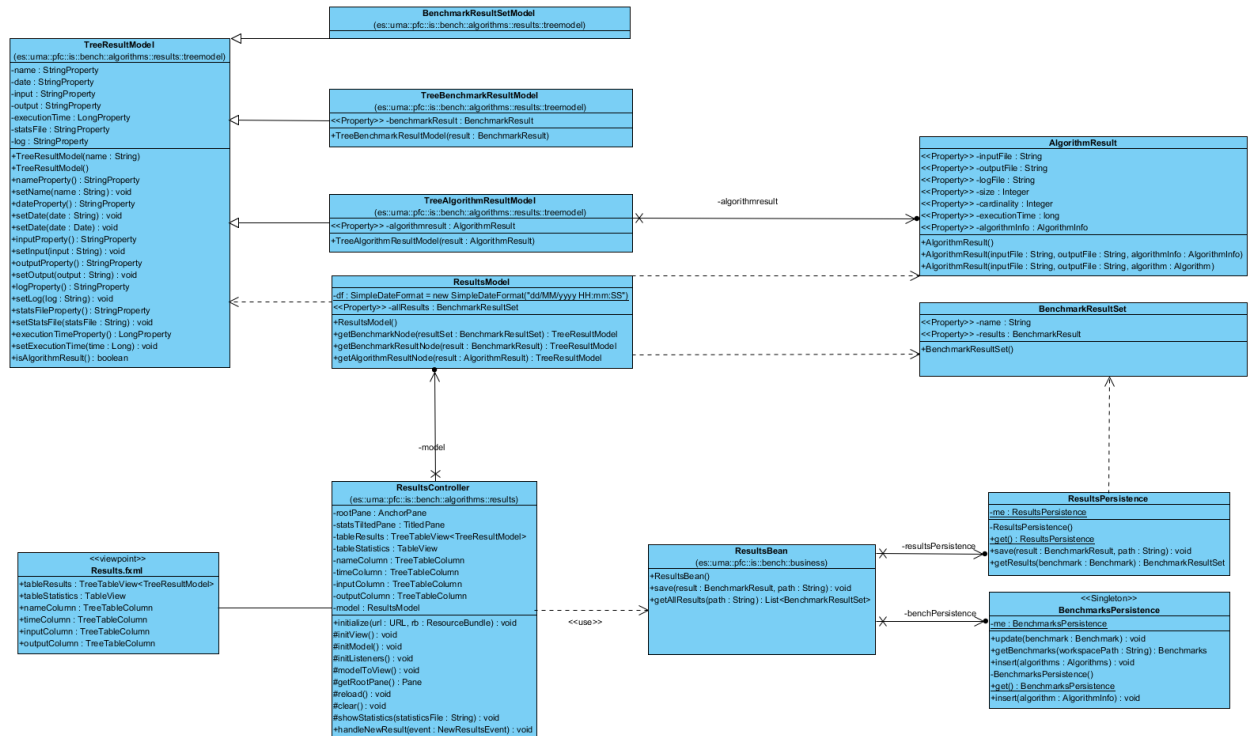
Operations

<u>public save (systems : java.util.List< fr.kbertet.lattice.ImplicationalSystem>, fileName : String) : void</u>		
Parameters	systems	
	Description	Implicational systems.
	Multiplicity	1
	Type	java.util.List < fr.kbertet.lattice.Implicational System>
	Direction	inout
	fileName	
	Description	File names prefix.
	Multiplicity	Unspecified
	Type	● String
	Direction	inout
Description	Java Detail	
Exceptions	java.io.IOException if read / write error occur.	













N/A

<u>protected checkFile (fileName : String) : void</u>		
Parameters	fileName	
	Description	File name.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Checks if the file with name fileName parameter value exists. If no exists, this will be created.	
Exceptions	java.io.IOException if read / write error occur.	

1.5.7. Resultados



Summary

Name	Description
 Results.fxml	Results view.
 ResultsController	Results view controller.
 ResultsModel	Results view model.
 TreeResultModel	Results tree table view model. Represents a row of the results table.
 TreeBenchmarkResultModel	Model for benchmark result node.
 TreeAlgorithmResultModel	Model for algorithm result nodes.
 BenchmarkResultSetModel	Benchmark result set tree node model.
 AlgorithmResult	Algorithm result info.
 BenchmarkResultSet	Collection of benchmark results.
 ResultsBean	Results bean.
 ResultsPersistence	Class for the read and write results to/from files.
 BenchmarksPersistence	Persists the benchmarks into an XML file entities using JAXB. The details can be found in the <i>Registrar Benchmarks</i> diagram class.

Details



Results.fxml

Name	Value
Visibility	public
Stereotypes	viewpoint

Attributes

public tableResults : javafx.scene.control.TreeTableView			
Description	Table which contains the results registered in the current workspace, with a tree hierarchy.		
Type	javafx.scene.control.TreeTableView		
Getter	false	Setter	false
Multiplicity	1		

public tableStatistics : TableView			
Description	Table which contains the statistics of the results registered in the current workspace.		
Type	javafx.scene.control.TableView		
Getter	false	Setter	false
Multiplicity	1		



ResultsController

Name	Value
Description	Results view controller.
Visibility	public

Attributes

private rootPane : javafx.scene.layout.AnchorPane	
Description	Root pane.

Type	javafx.scene.layout.AnchorPane		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private statsTitledPane : javafx.scene.control.TitledPane

Description	Pane which contains the statistics table.		
Type	javafx.scene.control.TitledPane		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private tableResults : javafx.scene.control.TreeTableView

Description	Table which contains the benchmarks execution results in a tree hirerchachy.		
Type	javafx.scene.control.TreeTableView		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private tableStatistics : javafx.scene.control.TableView

Type	javafx.scene.control.TableView		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		


private nameColumn : javafx.scene.control.TreeTableColumn

Type	javafx.scene.control.TreeTableColumn		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private timeColumn : javafx.scene.control.TreeTableColumn			
Type	javafx.scene.control.TreeTableColumn		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private inputColumn : javafx.scene.control.TreeTableColumn			
Type	javafx.scene.control.TreeTableColumn		
Getter	false	Setter	false
Derived	false		
Multiplicity	1		
Stereotypes	FXML		

private outputColumn : javafx.scene.control.TreeTableColumn			
Type	javafx.scene.control.TreeTableColumn		
Getter	false	Setter	false
Multiplicity	1		
Stereotypes	FXML		

private model : ResultsModel			
Description	Results view model.		
Type	 ResultsModel		
Getter	false	Setter	false
Multiplicity	1		

Operations

protected getRootPane () : javafx.scene.layout.Pane	
Description	Root pane.
Return Type Description	Pane.

public initialize (url : java.net.URL, rb : java.util.ResourceBundle) : void		
	url	
	Description	URL of the view.
	Multiplicity	1
	Type	java.net.URL
	Direction	inout
	rb	
	Description	Resource bundle.
	Multiplicity	1
	Type	java.util.ResourceBundle
	Direction	inout
Description	Initializes the view.	

protected initView () : void	
Description	Initializes the results table.
Upper	1
Exceptions	IOException

protected initModel () : void	
Description	Initializes the model.

protected initListeners () : void	
Description	Initializes the listeners.

protected modelToView () : void	
Description	Loads the model data into the results table.

protected reload () : void	
Description	Reloads the view and the model.

protected clear () : void	
Description	Clear the tables.

protected showStatistics (statisticsFile : String) : void		
Parameters	statisticsFile	
	Description	Statistics file path.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Shows the result statistics into a table.	

public handleNewResult (event : NewResultsEvent) : void		
Parameters	event	
	Description	Event.
	Multiplicity	1
	Type	■ NewResultsEvent
	Direction	inout
Description	Handles the NewResultsEvent published by Eventbus and reload the view and the model.	




ResultsModel

Name	Value
Description	Results view model.
Visibility	public


Attributes

<u>private df : java.text.SimpleDateFormat</u>			
Description	Date formatter for date column values.		
Initial Value	new SimpleDateFormat("dd/MM/yyyy HH:mm:ss")		
Type	java.text.SimpleDateFormat		
Getter	false	Setter	false
Multiplicity	1		


private allResults : BenchmarkResultSet			
Description	All results.		
Stereotypes	Property		
Type	 BenchmarkResultSet		
Getter	true	Setter	true
Multiplicity	0..*		

Operations

public ResultsModel ()	
Description	Constructor.

public getBenchmarkNode (resultSet : BenchmarkResultSet) : TreeResultModel		
Parameters	resultSet	
	Description	Benchmark result set.
	Multiplicity	1
	Type	 BenchmarkResultSet
	Direction	inout
Description	Returns a Benchmark node with a result set values passed by parameter.	
Return Type Description	Benchmark node.	

public getBenchmarkResultNode (result : es.uma.pfc.is.bench.domain.BenchmarkResult) : TreeResultModel		
Parameters	result	
	Description	Benchmark result.
	Multiplicity	1
	Type	es.uma.pfc.is.bench.domain.BenchmarkResult
	Direction	inout
Description	Creates a tree node with a benchmark result values.	
Return Description Type	A tree node with a benchmark result values.	

public getAlgorithmResultNode (result : AlgorithmResult) : TreeResultModel		
Parameters	result	
	Description	Algorithm execution result.
	Multiplicity	1
	Type	 AlgorithmResult
	Direction	inout
Description	Creates a tree node with an algorithm execution result values.	
Return Type Description	A tree node with a algorithm result values.	

TreeResultModel

Name	Value
Description	Results tree table view model. Represents a row of the results table.
Visibility	public

Attributes

private name : javafx.beans.property.StringProperty			
Description	Name column binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private date : javafx.beans.property.StringProperty			
Description	Date column binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private input : javafx.beans.property.StringProperty			
Description	Input column binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private output : javafx.beans.property.StringProperty			
Description	Output column binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private executionTime : javafx.beans.property.LongProperty			
Description	Execution time column binding property.		
Type	javafx.beans.property.LongProperty		
Getter	false	Setter	false
Multiplicity	1		

private statsFile : javafx.beans.property.StringProperty			
Description	Statistics file binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

private log : javafx.beans.property.StringProperty			
Description	Log file binding property.		
Type	javafx.beans.property.StringProperty		
Getter	false	Setter	false
Multiplicity	1		

Operations

public TreeResultModel (name : String)		
Parameters	name	
	Description	Name.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Constructor.	

public TreeResultModel ()	
Description	Constructor.

public nameProperty () : javafx.beans.property.StringProperty	
Description	Returns the name column binding property.
Return Type Description	The name column binding property.

public setName (name : String) : void		
Parameters	name	
	Description	Name.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Establishes the name column value.	

public dateProperty () : javafx.beans.property.StringProperty	
Description	Returns the date column binding property.
Return Type Description	The date column binding property.

public inputProperty () : javafx.beans.property.StringProperty	
Description	Returns the input column binding property.
Return Type Description	The input column binding property..

public setInput (input : String) : void		
Parameters	output	
	Description	Input.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Establishes the input column value.	

public setDate (date : String) : void		
Parameters	date	
	Description	Date string.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Establishes the date column value from a string value.	

public setDate (date : java.util.Date) : void		
Parameters	date	
	Description	Date.
	Multiplicity	1
	Type	java.util.Date
	Direction	inout
Description	Establishes the date with the format "dd/MM/yyyy HH:mm:ss".	

public outputProperty () : javafx.beans.property.StringProperty	
Description	Returns the output column binding property.
Return Type Description	The output column binding property..

public logProperty () : javafx.beans.property.StringProperty	
Description	Returns the log binding property.
Return Type Description	The log binding property.

public setOutput (output : String) : void		
Parameters	output	
	Description	Output.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Establishes the output column value	

public setLog (log : String) : void		
Parameters	log	
	Description	Log path.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Establishes the execution log path.	

public statsFileProperty () : javafx.beans.property.StringProperty	
Description	Returns the statistics file binding property.
Return Type Description	The statistics file binding property..

public isAlgorithmResult () : boolean	
Description	If the row is an algorithm result.
Return Type Description	true if is an algorithm result, false otherwise.

public setStatsFile (statsFile : String) : void		
Parameters	statsFile	
	Description	Statistics file path.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Establishes the statistics file path.	

public executionTimeProperty () : javafx.beans.property.LongProperty	
Description	Returns the execution time column binding property.
Return Type Description	The execution time column binding property..

public setExecutionTime (time : Long) : void		
Parameters	time	
	Description	Execution time.
	Multiplicity	1
	Type	Long
	Direction	inout
Description	Establishes the execution time.	



TreeBenchmarkResultModel

Name	Value
Description	Model for benchmark result node.
Visibility	public

Attributes

private benchmarkResult : BenchmarkResult			
Description	Benchmark result.		
Stereotypes	Property		
Type	BenchmarkResult		
Getter	true	Setter	false
Multiplicity	1		

Operations

public TreeBenchmarkResultModel (result : es.uma.pfc.is.bench.domain.BenchmarkResult)		
Parameters	result	
	Description	Benchmark result.
	Multiplicity	1
	Type	es.uma.pfc.is.bench.domain.BenchmarkResult
	Direction	inout
Description	Constructor. Initializes the node with a benchmark result values.	




TreeAlgorithmResultModel

Name	Value
Description	Model for algorithm result nodes.
Visibility	public

Attributes

private algorithmresult : AlgorithmResult			
Description	Algorithm result.		
Stereotypes	Property		
Type	AlgorithmResult		
Getter	true	Setter	false
Derived	false		
Multiplicity	1		

Operations

public TreeAlgorithmResultModel (result : AlgorithmResult)		
Parameters	result	
	Description	Algorihtm result.
	Multiplicity	1
	Type	 AlgorithmResult
	Direction	inout
Description	Constructor.	



BenchmarkResultSetModel

Name	Value
Description	Benchmark result set tree node model.
Visibility	public




BenchmarkResultSet


Name	Value
Description	Collection of benchmark results.
Visibility	public

Operations

public BenchmarkResultSet ()	
Description	Constructor.

Attributes

private name : String			
Description	Benchmark name.		
Stereotypes	Property		
Type	 String		
Getter	true	Setter	true
Multiplicity	1		


private results : BenchmarkResult			
Description	Benchmark results.		
Stereotypes	Property		
Type	 BenchmarkResult		
Getter	true	Setter	true
Multiplicity	0..*		



AlgorithmResult

Name	Value
Description	Algorithm result info.
Visibility	public

Attributes


private inputFile : String			
Description	Implicational system input file.		
Stereotypes	Property		
Type	 String		
Getter	true	Setter	false
Multiplicity	1..*		


private outputFile : String			
Description	Implicational system output file.		
Stereotypes	Property		
Type	● String		
Getter	true	Setter	false
Multiplicity	1		

private logFile : String			
Description	Log file.		
Stereotypes	Property		
Type	● String		
Getter	true	Setter	true
Multiplicity	1		

private size : Integer			
Description	Implicational System size.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	false
Multiplicity	1		



private cardinality : Integer			
Description	Implicational System cardinality.		
Stereotypes	Property		
Type	Integer		
Getter	true	Setter	false
Multiplicity	1		


private executionTime : long			
Description	Execution time.		
Stereotypes	Property		
Type	 long		
Getter	true	Setter	true
Multiplicity	1		




private algorithmInfo : AlgorithmInfo			
Description	Info of the executed algorithm.		
Stereotypes	Property		
Type	 AlgorithmInfo		
Getter	true	Setter	false
Multiplicity	1		

Operations

public AlgorithmResult ()	
Description	Constructor.

public AlgorithmResult (inputFile : String, outputFile : String, algorithm : Algorithm)		
Parameters	inputFile	
	Description	Input system file.
	Multiplicity	1
	Type	 String
	Direction	inout
	outputFile	
	Description	Output system file.
	Multiplicity	1
	Type	 String
	Direction	inout
	algorithm	
	Description	Algorithm.
	Multiplicity	1


public AlgorithmResult (inputFile : String, outputFile : String, algorithm : Algorithm)		
	Type	 Algorithm
	Direction	inout
Description	Constructor.	


public AlgorithmResult (inputFile : String, outputFile : String, algorithmInfo : AlgorithmInfo)		
Parameters	inputFile	
	Description	Input system file.
	Multiplicity	1
	Type	 String
	Direction	inout
	outputFile	
	Description	Output system file.
	Multiplicity	1
	Type	 String
	Direction	inout
	algorithmInfo	
	Description	Algorithm info.
	Multiplicity	1
	Type	 AlgorithmInfo
	Direction	inout
Description	Constructor.	

ResultsBean

Name	Value
Description	Logic for reads and insert benchmark results in files.
Visibility	public


Attributes

private benchPersistence : BenchmarksPersistence			
Description	Benchmarks persistence.		
Type	 BenchmarksPersistence		
Getter	false	Setter	false
Multiplicity	1		

private resultsPersistence : ResultsPersistence			
Description	Results persistence.		
Type	 ResultsPersistence		
Getter	false	Setter	false
Multiplicity	1		

Operations

public ResultsBean ()	
Description	Constructor.


public save (result : es.uma.pfc.is.bench.domain.BenchmarkResult, path : String) : void		
Parameters	result	
	Description	Benchmark result.
	Multiplicity	1
	Type	es.uma.pfc.is.bench.domain.BenchmarkResult
	Direction	inout
	path	
	Description	Directory of results.xml.
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Saves a benchmark result in results.xml file.	

public getAllResults (path : String) : java.util.List		
Parameters	path	
	Description	Path of registered benchmarks.
	Multiplicity	1
	Type	● String
	Direction	inout
Description	Returns all saved results of the registered benchmarks in the path.	
Return Type Description	Results saved.	

ResultsPersistence

Name	Value
Description	Class for the read and write results to/from files.
Visibility	public


Attributes


private me : ResultsPersistence			
Description	Single instance.		
Type	 ResultsPersistence		
Getter	false	Setter	false
Multiplicity	1		

Operations

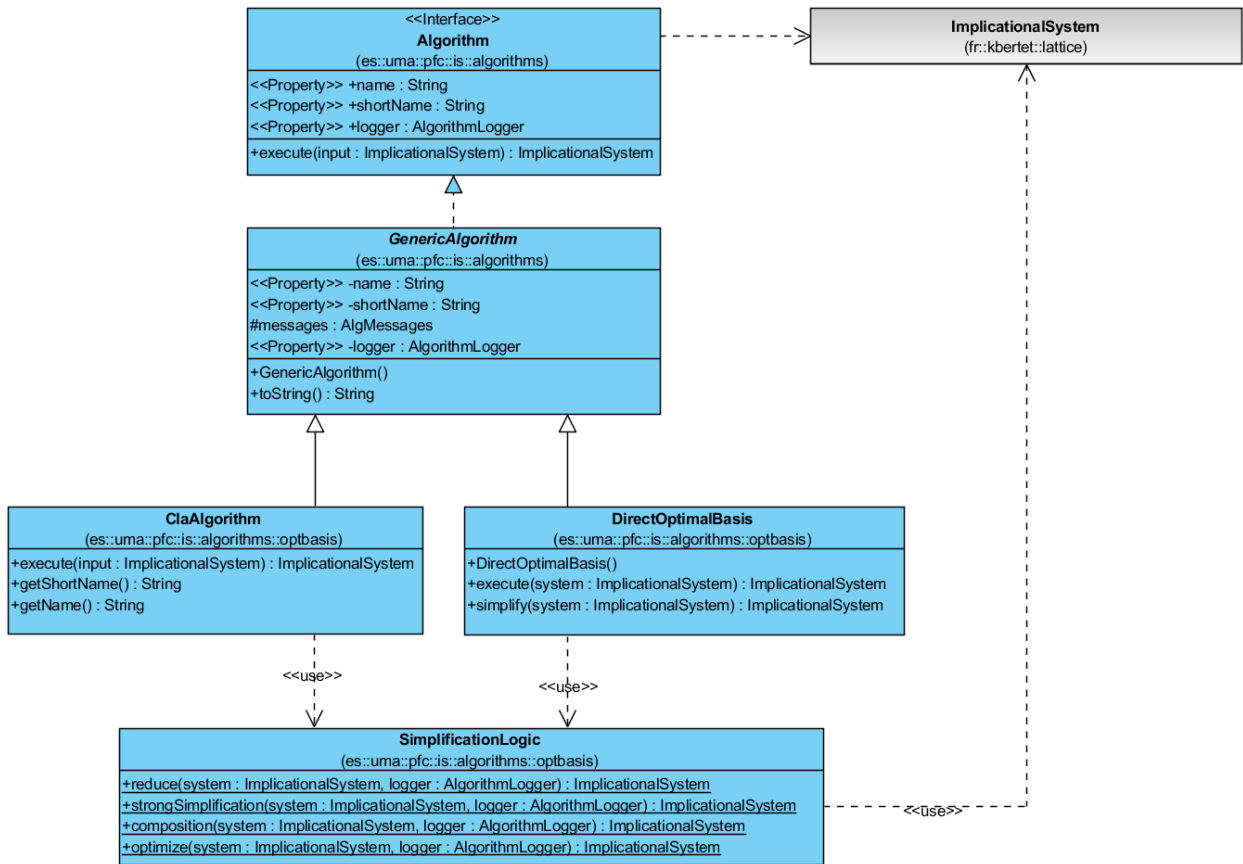
private ResultsPersistence ()	
Description	Constructor.

public get () : ResultsPersistence	
Description	Gets a single instance of ResultsPersistence.
Return Type Description	ResultsPersistence single instance.







public getResults (benchmark : Benchmark) : BenchmarkResultSet		
Parameters	benchmark	
	Description	Benchmark.
	Multiplicity	1
	Type	 Benchmark
	Direction	inout
Description	Returns a benchmark's saved results.	
Return Type Description	Benchmark's saved results. null if the results not exists.	

public save (result : es.uma.pfc.is.bench.domain.BenchmarkResult, path : String) : void		
Parameters	result	
	Description	Benchmark result to save.
	Multiplicity	1
	Type	es.uma.pfc.is.bench.domain.BenchmarkResult
	Direction	inout
	path	
	Description	Results file directory path.
	Multiplicity	1
	Type	 String
	Direction	inout
Description	Saves a benchmark result to the results.xml file in a directory passed by parameter.	

1.5.8. API Algoritmos



Summary

Name	Description
 Algorithm	Algorithm of implicational system basis computation.
 GenericAlgorithm	Generic algorithm which receive as input a file input and returns an implicational system.
 DirectOptimalBasis	AMIS Algorithm implementation.
 ClaAlgorithm	CLA Algorithm implementation.
 ImplicationalSystem	This class gives a representation for an implicational system (ImplicationalSystem), a set of rules. This class belong to java-lattices library.
 SimplificationLogic	Methods which implements the simplification logic rules.

Description

API for implementation of algorithms which can be executed by IS Bench.

Details



Algorithm

Name	Value
Description	Algorithm of implicational system basis computation. This interface must be implemented by the classes which will be executed by IS Bench.
Visibility	public

Attributes

public name : String			
Description	Algorithm name. It will be used as the algorithm string representation.		
Stereotypes	Property		
Type	● String		
Getter	true	Setter	true
Multiplicity	1		

public shortName : String			
Description	Algorithm short name. The short name will be used as the default name of the output files.		
Stereotypes	Property		
Type	● String		
Getter	true	Setter	true
Multiplicity	1		

public logger : es.uma.pfc.is.logging.AlgorithmLogger			
Description	Logger for trace generation.		
Stereotypes	Property		
Type	es.uma.pfc.is.logging.AlgorithmLogger		
Getter	true	Setter	false
Multiplicity	1		

Operations

public execute (input : fr.kbertet.lattice.ImplicationalSystem) : fr.kbertet.lattice.ImplicationalSystem .ImplicationalSystem		
Parameters	input	
	Description	Input implicational system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
Description	Executes the algorithm with the implicational system parameter as input.	
Return Type Description	Implicational System returned by the executed algorithm.	




GenericAlgorithm

Name	Value
Description	Generic algorithm which receive as input a file input and returns an implicational system.
Visibility	public

Attributes

private name : String			
Description	Name.		
Stereotypes	Property		
Type	String		
Getter	true	Setter	true
Multiplicity	1		

private shortName : String			
Description	Short name.		
Stereotypes	Property		
Type	String		
Getter	true	Setter	true
Multiplicity	1		

protected messages : AlgMessages			
Description	Translates messages to current language.		
Type	 AlgMessages		
Getter	false	Setter	false
Multiplicity	1		

private logger : es.uma.pfc.is.logging.AlgorithmLogger			
Description	Logger.		
Stereotypes	Property		
Type	es.uma.pfc.is.logging.AlgorithmLogger		
Getter	true	Setter	false
Multiplicity	1		

Operations


public GenericAlgorithm ()	
Description	Constructor.

protected setLogger (logger : es.uma.pfc.is.logging.AlgorithmLogger) : void		
Parameters	logger	
	Description	Logger.
	Multiplicity	1
	Type	es.uma.pfc.is.logging.AlgorithmLogger
	Direction	inout
Description	For testing usage.	

protected removeRule (system : fr.kbertet.lattice.ImplicationalSystem, rule : fr.kbertet.lattice.Rule) : void		
Parameters	system	
	Description	Implicational system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	rule	
	Description	Rule.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
Description	Removes rules for implicational system, and print trace in the log.	

protected addRule (system : fr.kbertet.lattice.ImplicationalSystem, rule : fr.kbertet.lattice.Rule) : void		
Parameters	system	
	Description	Implicational system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	rule	
	Description	Rule.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
Description	Adds rules for implicational system, and print trace in the log.	
Return Type Description	Implicational system with a rule added.	

protected addRuleAndElements (system : fr.kbertet.lattice.ImplicationalSystem, rule : fr.kbertet.lattice.Rule) : ImplicationalSystem		
Parameters	system	
	Description	Implicational system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	rule	
	Description	Rule.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
Description	Adds rules for implicational system, and print trace in the log.	
Return Type Description	Implicational system with a rule and its elements added.	

protected addRuleAndElements (system : fr.kbertet.lattice.ImplicationalSystem, rule : fr.kbertet.lattice.Rule, trace : boolean) : fr.kbertet.lattice.ImplicationalSystem		
Parameters	system	
	Description	Implicational system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	rule	
	Description	Rule.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
	trace	
	Description	If print trace in the log.
	Multiplicity	1
	Type	 boolean
	Direction	inout
Description	Adds rules for implicational system and its elements, and print trace in the log if the trace parameter is true.	

protected history (message : String, args : Object) : void		
Parameters	message	
	Description	Message.
	Multiplicity	1
	Type	String
	Direction	inout
	args	
	Description	Message arguments.
	Multiplicity	0..*
	Type Modifier	...
	Type	Object
	Direction	inout
Description	Prints a message with the arguments, to the log.	

protected replaceRule (system : fr.kbertet.lattice.ImplicationalSystem, rule1 : fr.kbertet.lattice.Rule, rule2 : fr.kbertet.lattice.Rule) : void		
Parameters	system	
	Description	Implicational system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	rule1	
	Description	Rule to replace.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
	rule2	
	Description	New rule.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout

Description	Replace a rule by other for implicational system, and print trace in the history.
-------------	---

public toString () : String	
Description	Algorithm string representation. By default, is the name property value.
Return Type Description	Name.





DirectOptimalBasis

Name	Value
Description	Direct Optimal Basis algorithm implementation.
Visibility	public

Operations

public DirectOptimalBasis ()	
Description	Constructor.

public execute (system : ImplicationalSystem) : ImplicationalSystem		
Parameters	system	
	Description	Input system.
	Multiplicity	1
	Type	 ImplicationalSystem
	Direction	inout
Description	Executes the Direct Optimal Basis algorithm.	

public simplify (system : ImplicationalSystem) : ImplicationalSystem		
Parameters	system	
	Description	Reduced system.
	Multiplicity	1
	Type	 ImplicationalSystem
	Direction	inout
Description	Generation of IS simplificated by simplification(left+right+composition) of reduced IS	
Return Type Description	Simplified system.	
Query	false	

protected printInit (inputSystem : fr.kbertet.lattice.ImplicationalSystem) : void		
Parameters	inputSystem	
	Description	Implicactional System.
	Multiplicity	Unspecified
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
Description	Prints de initial arguments.	


protected printResult (resultSystem : fr.kbertet.lattice.ImplicationalSystem) : void		
Parameters	resultSystem	
	Description	Implicational System.
	Multiplicity	Unspecified
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
Description	Prints the results.	



ClaAlgorithm

Name	Value
Description	CLA Algorithm implementation.
Visibility	public

Operations

public execute (input : ImplicationalSystem) : ImplicationalSystem		
Parameters	input	
	Description	Input implicational system.
	Multiplicity	1
	Type	 ImplicationalSystem
	Direction	inout
Description	Executes the CLA algorithm for computation of the input implicational system direct basis. Uses the SimplificationLogic class methods.	

public getShortName () : String	
Description	Short name.

public getName () : String	
Description	Name.



SimplificationLogic

Name	Value
Description	Methods which implements the simplification logic rules.
Visibility	public

Operations

public fragmentationEquivalency (implication : fr.kbertet.lattice.Rule) : fr.kbertet.lattice.Rule		
Parameters	implication	
	Description	Implication.
	Multiplicity	1
	Type	Rule
	Direction	inout
Description	Implements the Fragmentation Equivalency rule: [FrEq]: {A -> B} = {A -> B-A}.	
Return Type Description	Equivalent simplified implication.	


<u>public compositionEquivalency (rule1 : fr.kbertet.lattice.Rule,</u> <u>rule2 : fr.kbertet.lattice.Rule)</u> <u>: java.util.List< fr.kbertet.lattice.Rule></u>		
Parameters	rule1	
	Description	Implication.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
	rule2	
	Description	Implication.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
Description	Given two implications: $\{A \rightarrow B, A \rightarrow C\} \Rightarrow \{A \rightarrow BC\}$	
Return Type Description	If the composition rule can be applied, returns a list with once implication from the rule application. If the composition rule can't be applied, returns a list which contains the rules passed as parameters.	
Exceptions	java.lang.NullPointerException Si alguna de las implicaciones es nula.	


<u>public rightSimplificationEq (rule1 : fr.kbertet.lattice.Rule,</u> <u>rule2 : fr.kbertet.lattice. Rule)</u> <u>: java.util.List< fr.kbertet.lattice.></u>		
Parameters	rule1	
	Description	Implication.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
	rule2	
	Description	Implication.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
Description	Implements the simplification rule for two implications.	


<u>public rightSimplificationEq (rule1 : fr.kbertet.lattice.Rule,</u> <u>rule2 : fr.kbertet.lattice. Rule)</u> <u>: java.util.List< fr.kbertet.lattice.></u>	
	[SiEq]: if (A intersection B is empty) and (A subset of C) then $\{A \rightarrow B, C \rightarrow D\} == \{A \rightarrow B, C-B \rightarrow D-B\}$
Return Type Description	Simplified implications.
Exceptions	java.lang.NullPointerException if any rule is null.


<u>public compositionEquivalency (is : fr.kbertet.lattice.ImplicationalSystem,</u> <u>rule1 : fr.kbertet.lattice.Rule,</u> <u>rule2 : fr.kbertet.lattice.Rule) :</u> <u>fr.kbertet.lattice.ImplicationalSystem</u>		
Parameters	is	
	Description	Implicational system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	rule1	
	Description	Implication.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
	rule2	
	Description	Implication.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
Description	rule1 and rule2 are implications from is implicational system parameter. If the composition rule can be applied to rule1 and rule2, removes these rules from the implicational system and adds the new.	

<u>public strongSimplificationEq (rule1 : fr.kbertet.lattice.Rule,</u> <u>rule2 : fr.kbertet.lattice.Rule)</u> <u>: fr.kbertet.lattice.Rule</u>		
Parameters	rule1	
	Description	Implication.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
	rule2	
	Description	Implicación.
	Multiplicity	1
	Type	fr.kbertet.lattice.Rule
	Direction	inout
Description	If (B intersection C) not is empty and (D \ A union B)) neither, returns the new implication $AC - B \rightarrow D - (AB)$.	
Return Type Description	Returns a new implication if Strong Simplification can be applied, null otherwise.	

<u>public reduce (system : ImplicationalSystem, logger : AlgorithmLogger)</u> <u>: fr.kbertet.lattice.ImplicationalSystem</u>		
Parameters	system	
	Description	Implicational system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	logger	
	Description	Logger.
	Multiplicity	1
	Type	 AlgorithmLogger
	Direction	inout
Description	Gets a reduced system. An implicational system is reduced, if $A \rightarrow B \Rightarrow (B \text{ not empty})$ AND $(A \text{ intersection } B \text{ is empty})$ for all A, B in S.	
Return Type Description	Reduced implicational system.	

<u>public strongSimplification (system : fr.kbertet.lattice.ImplicationalSystem,</u> <u>logger : AlgorithmLogger)</u> <u>: fr.kbertet.lattice.ImplicationalSystem</u>		
Parameters	system	
	Description	Simplified system.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	logger	
	Description	Logger.
	Multiplicity	1
	Type	 AlgorithmLogger
	Direction	inout
Description	Generation of IS by completion of simplified IS --> Strong Simplification.	
Return Type Description	Strong simplified system.	
Exceptions	java.lang.NullPointerException if system is null.	

<u>public composition (system : fr.kbertet.lattice.ImplicationalSystem,</u> <u>logger : AlgorithmLogger)</u> <u>: fr.kbertet.lattice. ImplicationalSystem</u>		
Parameters	system	
	Description	Implicational System.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	logger	
	Descriptoin	Loger
	Multiplicity	1
	Type	 AlgorithmLogger
	Direction	inout
Description	Composition of implications of a system.	
Return Type Description	Implicational System with composition applied.	

<u>public optimize (system : fr.kbertet.lattice.ImplicationalSystem,</u> <u>logger : AlgorithmLogger) : fr.kbertet.lattice.ImplicationalSystem</u>		
Parameters	system	
	Description	Simplified Implicational System.
	Multiplicity	1
	Type	fr.kbertet.lattice.ImplicationalSystem
	Direction	inout
	logger	
	Description	Logger.
	Multiplicity	1
	Type	 AlgorithmLogger
	Direction	inout
Description	Generation of optimized IS.	
Return Type Description	Optimized system.	

1.6. Diagramas de secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

En esta sección, se presentan los diagramas de secuencia de los escenarios más relevantes.

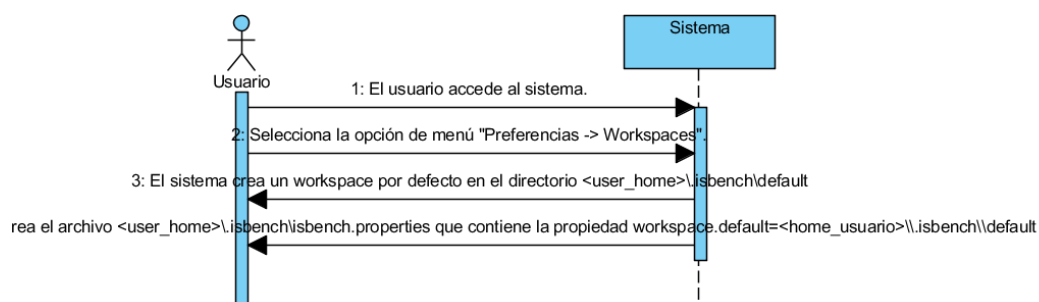


Figura 1.3: Primer acceso al sistema

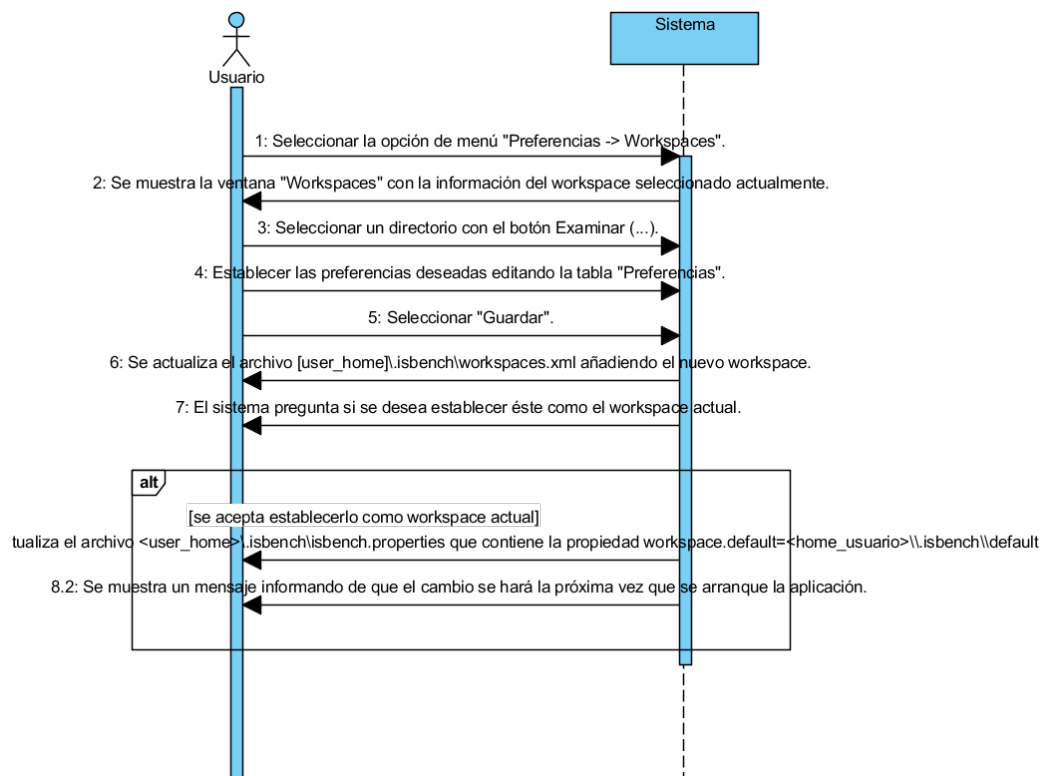


Figura 1.4: Crear un workspace

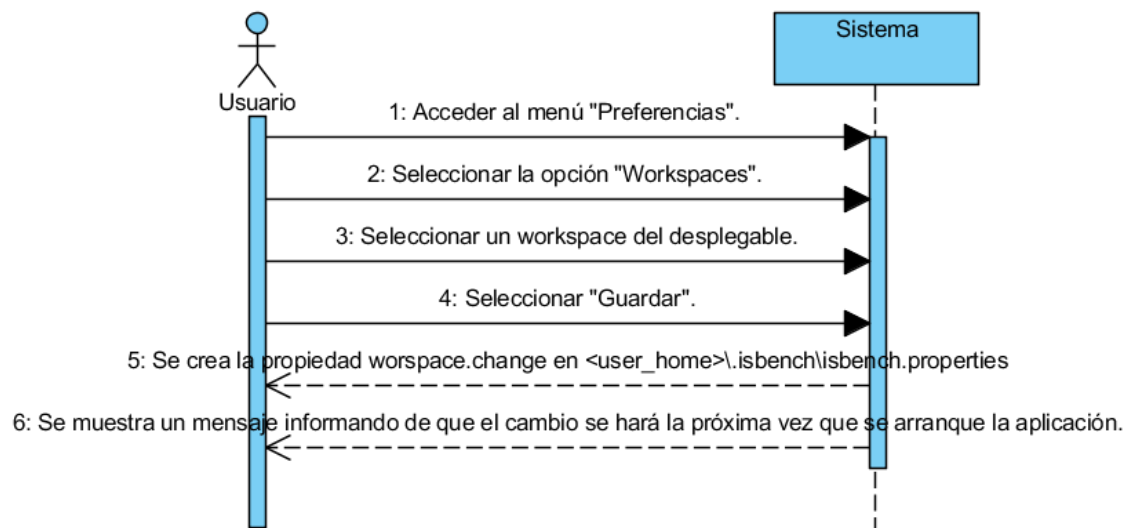


Figura 1.5: Cambiar de workspace

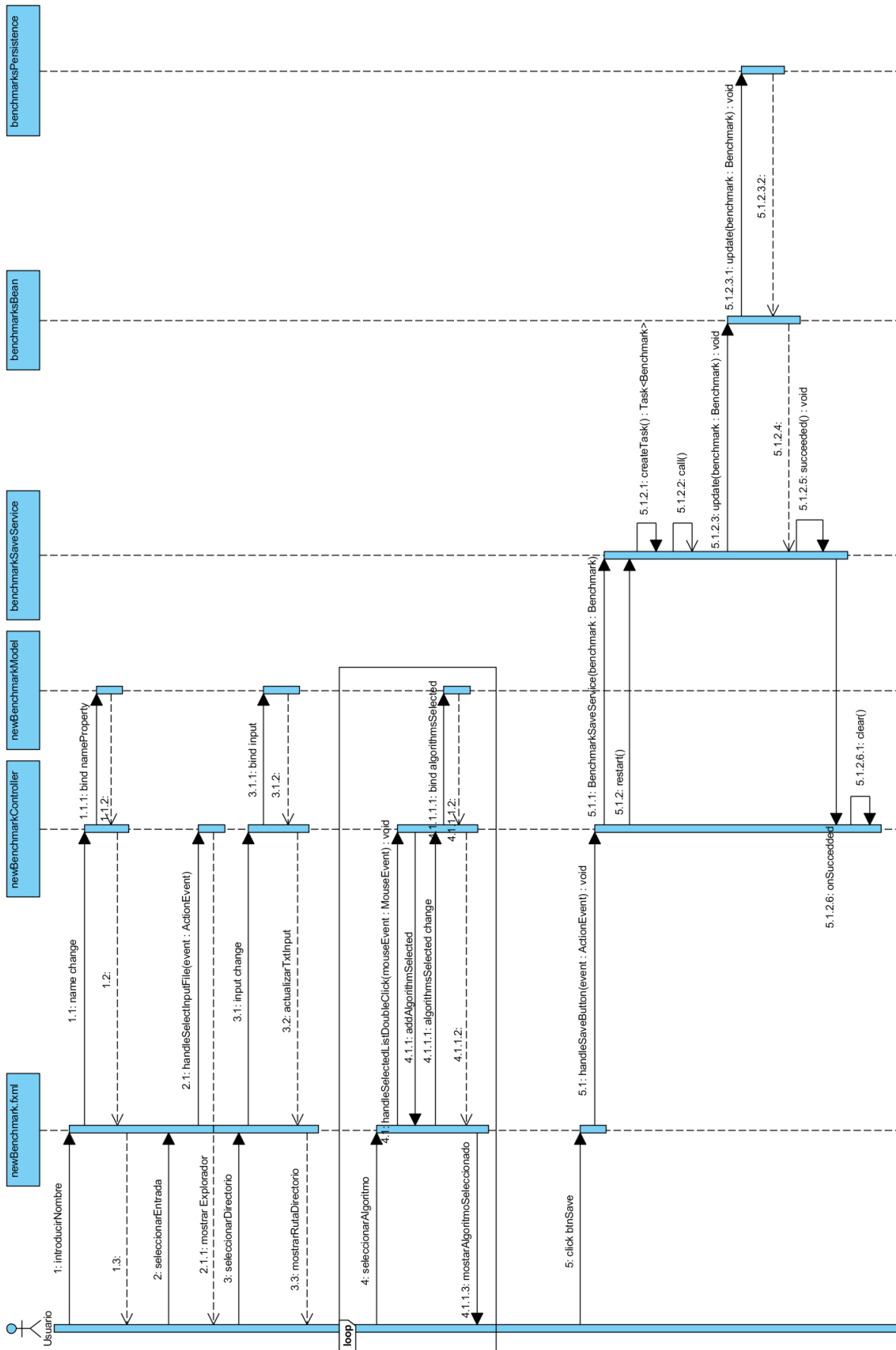


Figura 1.6: Registrar Benchmark

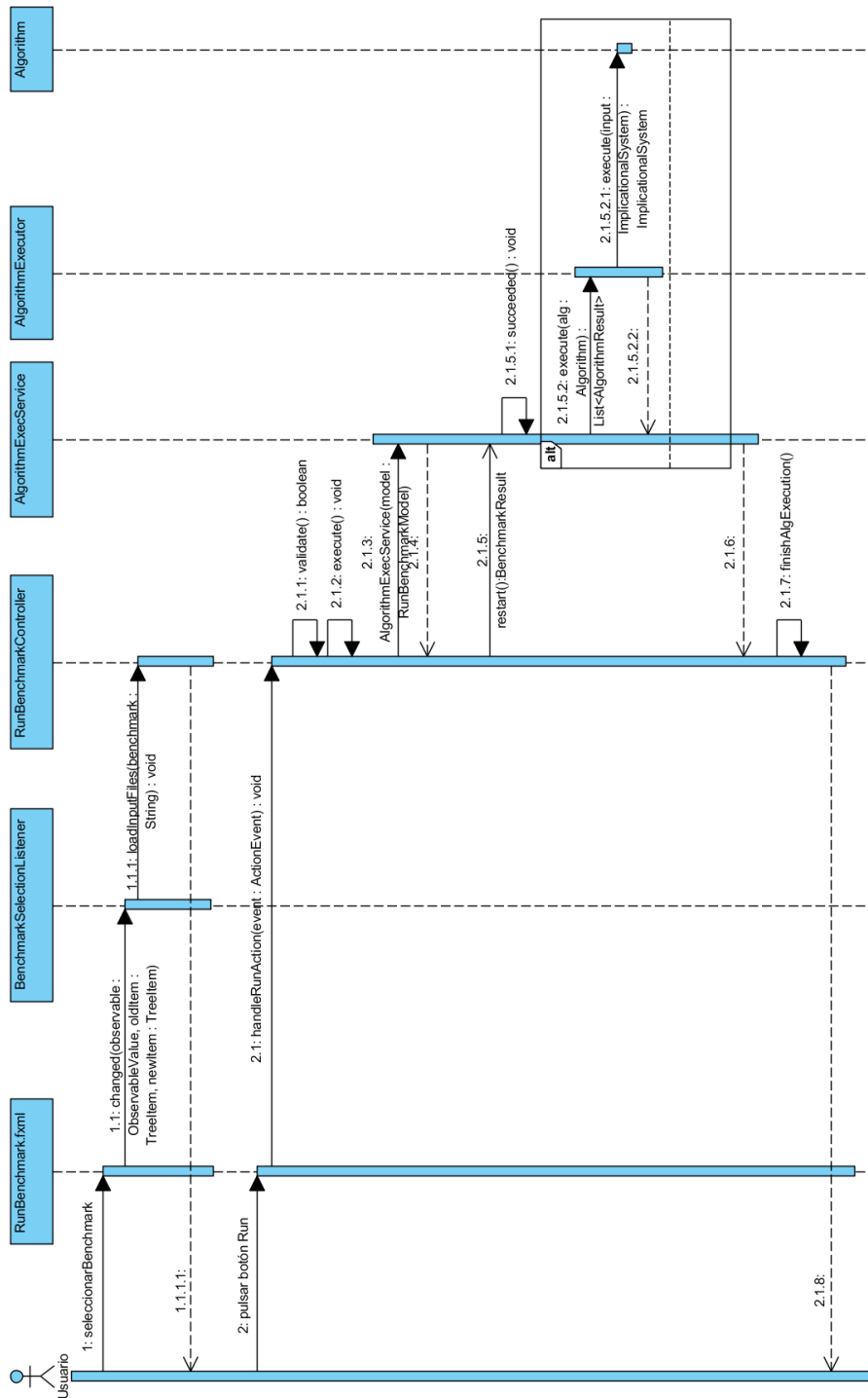


Figura 1.7: Ejecutar Benchmark