



API para el desarrollo de algoritmos interactivos en ingeniería del software basada en búsqueda

AURORA RAMÍREZ, JOSÉ RAÚL ROMERO, SEBASTIÁN VENTURA
UNIVERSIDAD DE CÓRDOBA

*XXIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD)
Sevilla (España), 17-19 de septiembre de 2018*

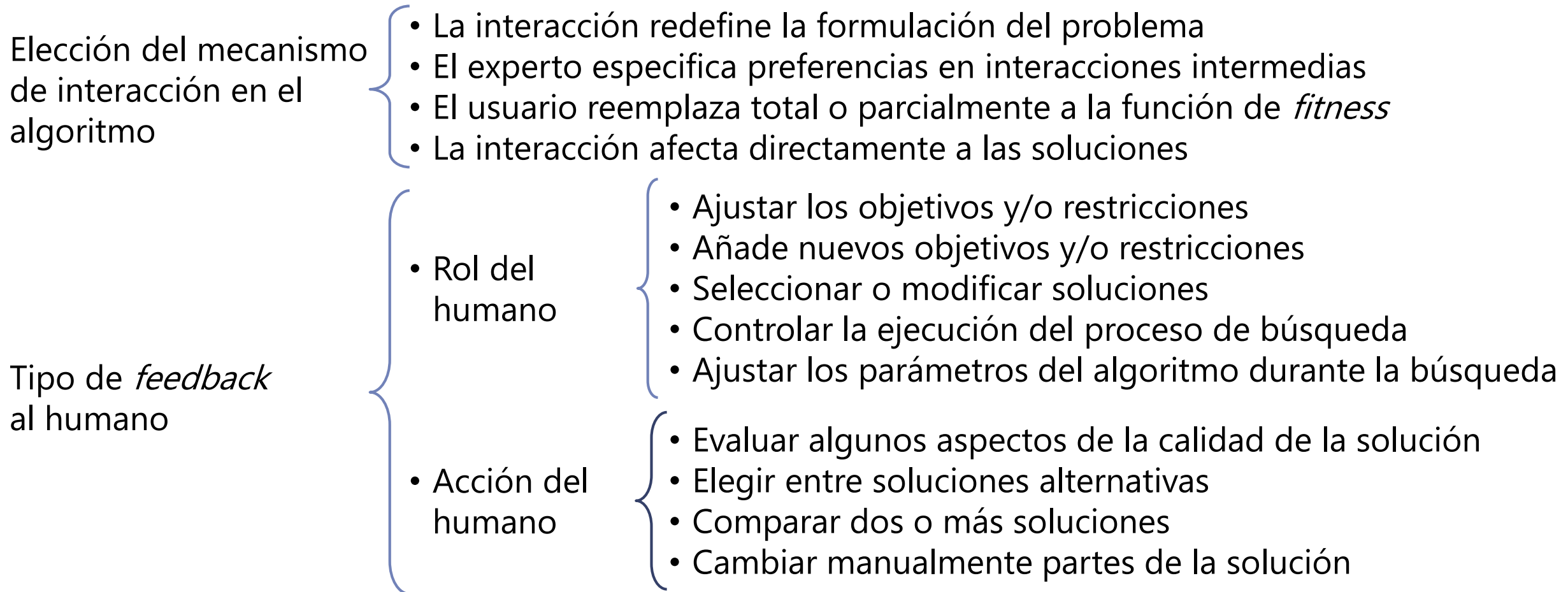
Índice de contenido

1. Introducción
2. Interactividad en SBSE
3. Especificación de la API
4. Ejemplo práctico de uso
5. Conclusiones

Introducción

- Ingeniería del software basada en **búsqueda interactiva (iSBSE)**
 - Hacer partícipe al experto durante el proceso de búsqueda
 - Creatividad, intuición, experiencia... son difíciles de transmitir
- Necesidad de un **mecanismo de interacción** efectivo y atractivo
 - Específico para el problema
 - Manejar la fatiga o la incertidumbre
- API de soporte a la programación en **Java**
 - Que tenga en cuenta los factores propios de iSBSE
 - Independiente de las librerías para la codificación de algoritmos

Interactividad en SBSE



Interactividad en SBSE

Características de la intervención del humano

• Ajuste del tiempo de intervención

- En cada iteración
- Cada N iteraciones
- Entre 2 ejecuciones del mismo o distinto algoritmo
- Bajo demanda del humano
- Adaptativo (según criterios fijados por el algoritmo)

• Visualización de las soluciones

- Decidir el número de soluciones a mostrar
- Decidir el nivel de detalle de la información ofrecida
- Establecer criterio de selección de soluciones

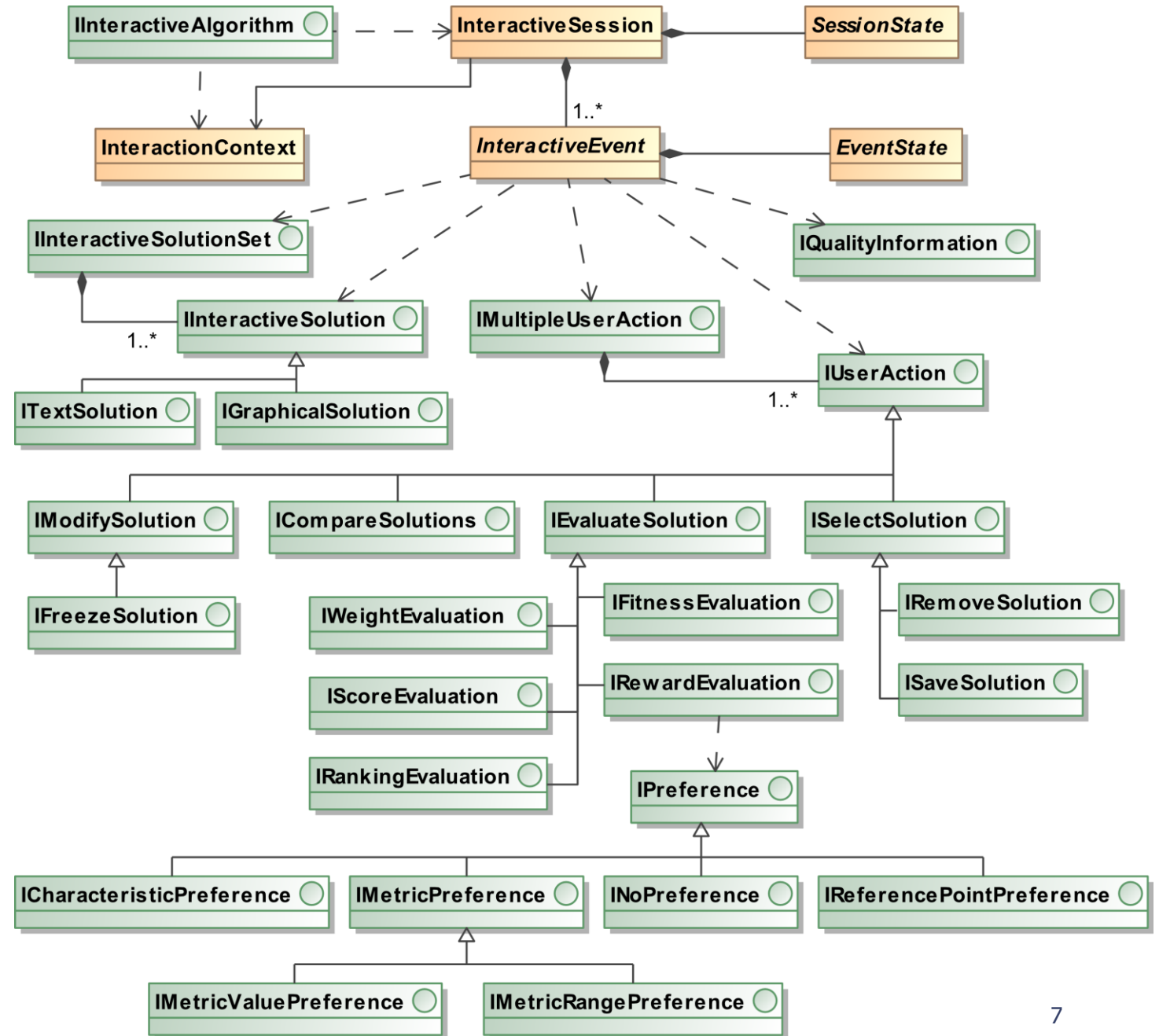
Interactividad en SBSE

Naturaleza de la evaluación del humano

- Proporcionar valor de *fitness*
- Asignar pesos a los objetivos (propuestas multi-objetivo)
- Proporcionar puntuaciones a las soluciones
- Ordenar soluciones en un ranking
- Premiar/penalizar la calidad de una solución

Influencia de la opinión del humano en el algoritmo

- Duración de la vida de la opinión (iteración, ejecución, múltiples ejecuciones)
- Posibilidad de modificar la opinión suministrada conforme avanza la búsqueda



Especificación de la API

 Sesiones y eventos

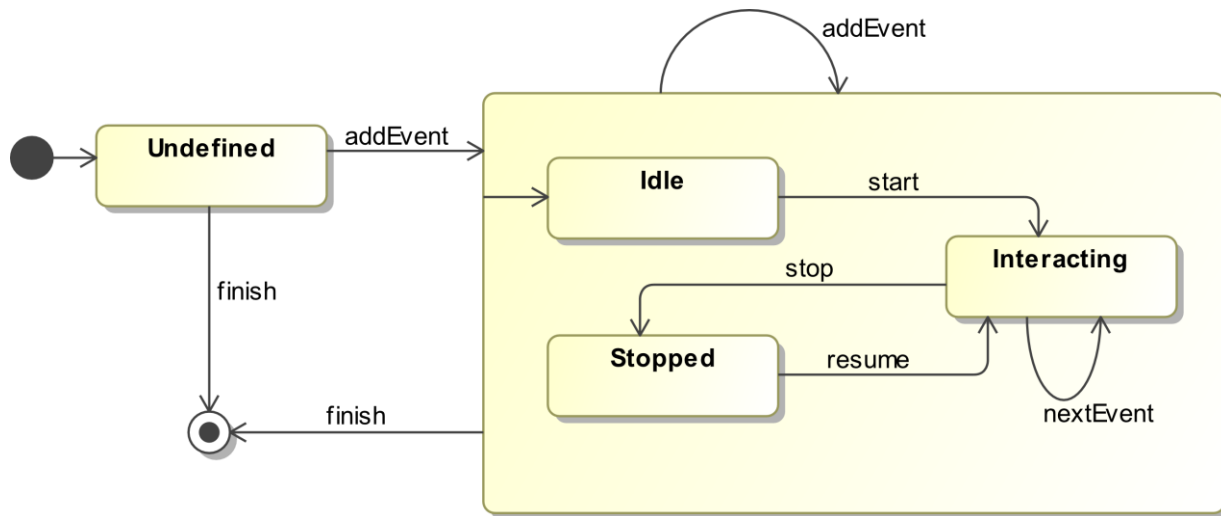
Soluciones 

 Acciones

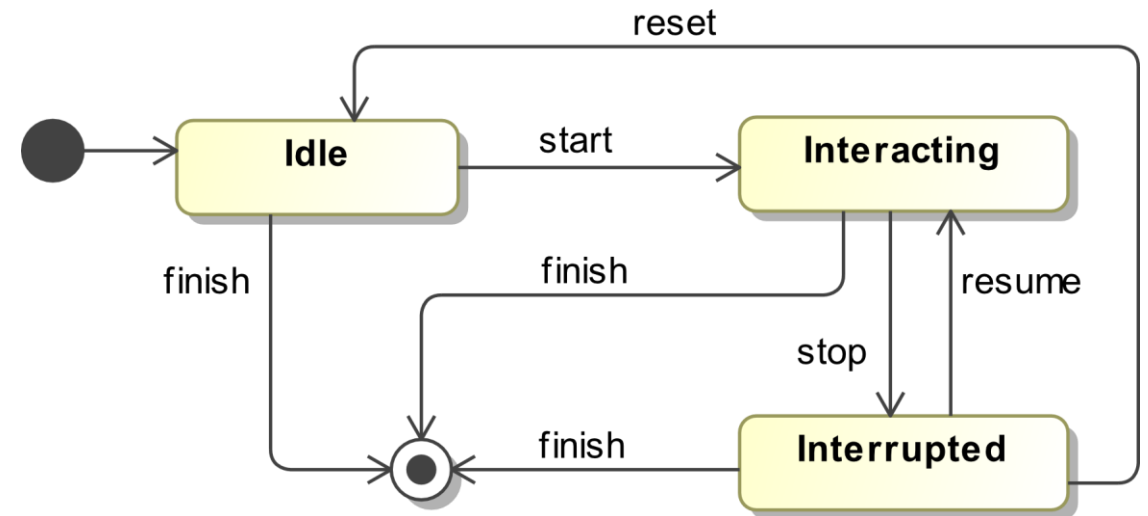
Evaluación 

Especificación de la API

SESIÓN INTERACTIVA



EVENTO INTERACTIVO



Especificación de la API



Soluciones

- Una / varias
- Texto / imagen
- Almacenar información cualitativa



Acciones

- Simple / múltiple
- Modificar partes
- Tres tipos de comparación
- Selección para eliminar o guardar



Evaluación

- Cuantitativa: fitness, ranking, pesos
- Cualitativa: premiar o penalizar según preferencias

Ejemplo práctico de uso

```
public class InteractiveDiscoveryEvent extends InteractiveEvent
    implements IXMISolution, IQualityInformation, IMultipleUserAction {

    protected String xmiSolutionDiagram; // Solución a mostrar
    protected double [][] designMetrics; // Medidas de diseño por componente
    protected double [] searchMetrics; // Medidas sobre la búsqueda
    protected List<IUserAction> actions; // Lista de acciones

    public String getDescription() {
        String description =
            LanguagePropertyReader.getInstance().getProperty("event.description",
                this.context.getLanguage());
        return description;
    }
    ...
}
```

```

...
// Métodos de la interfaz IXMISolution
public boolean isSolutionModifiable() {
    return false;
}
public InteractiveSolutionType getSolutionType() {
    return InteractiveSolutionType.COMP_DIAGRAM;
}
...
// Métodos de la interfaz IMultipleUserAction
public IUserAction getUserActionByType(InteractiveActionType type) {
    for(IUserAction action: this.actions) {
        if(action.getActionType() == type) {
            ...
        }
    }
    return null;
}
public void clearAllActions() {
    this.actions.forEach(a -> a.clearAction());
}
...
}

```

Ejemplo práctico de uso

```
public class FreezeComponentAction implements IFreezeSolution {
    protected int indexFrozenComponent; // Índice del componente seleccionado

    public void clearAction() {
        this.indexFrozenComponent = -1;
    }

    public boolean isFrozenSolution() {
        return (this.indexFrozenComponent != -1);
    }

    public int getNumberFreezableElements() {
        return 1;
    }

    ...
}
```

```

public class BestComponent implements ICharacteristicPreference {
    protected int numberOfComponents;    // Número de componentes
    protected int index;                // Índice del componente seleccionado

    public int getNumberOfOptions() {
        return this.numberOfComponents;
    }
    public boolean isMultipleOptionAllowed() {
        return false;
    }
    public void selectOption(int index) {
        this.index = index;
    }
    public boolean isOptionSelected(int index) {
        return (index == this.index);
    }
    ...
}

```

Conclusiones

- ✓ Promover la adopción de **técnicas interactivas** en SBSE
 - ✓ Facilitar el **diseño y desarrollo** de mecanismos de interacción
 - ✓ **Independizar** la interacción del algoritmo y la interfaz gráfica
-
- + Extender al **resto de factores** identificados en iSBSE
 - + **Integrar** con los frameworks habituales en SBSE

¡Gracias! ¿Preguntas?



UNIVERSIDAD DE CÓRDOBA

Aurora Ramírez

Email: aramirez@uco.es

Web: <http://www.uco.es/users/aramirez>

Twitter: @aurora_rq

API para el desarrollo de algoritmos interactivos en ingeniería del software basada en búsqueda

XXIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'18)