Identificación de Componentes en Arquitecturas Software Mediante Programación Evolutiva

A. Ramírez, J.R. Romero, S. Ventura

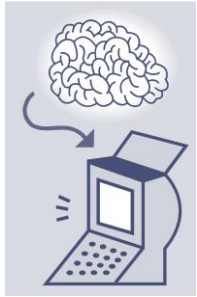
Dpto. de Informática y Análisis Numérico. Universidad de Córdoba, España.

XVIII Jornadas en Ingeniería del Software y Bases de Datos (JISBD 2013).
Madrid (España). 17-20 de septiembre de 2013.

Índice de contenido

Introducción

- Importancia del análisis arquitectónico
  - Concepción de la funcionalidad
  - Modificar o aumentar las funcionalidades que ofrece actualmente
  - Calidad
  - Experiencia y habilidades humanas
- Recuperación de la arquitectura del software
  - Identificación de bloques constituyentes y sus relaciones
  - Abstracción de modelos de análisis (generalmente, representados como diagramas de clases)
  - Automatización mediante técnicas de Inteligencia Artificial



Introducción

¿Qué es *Search-Based Software Engineering*?

Aplicación de técnicas de optimización y búsqueda propias de la Inteligencia Artificial a tareas de la Ingeniería del Software

Reformula problemas de la Ingeniería del Software para convertirlos en problemas de búsqueda y optimización

La Ingeniería del Software aporta *problemas*

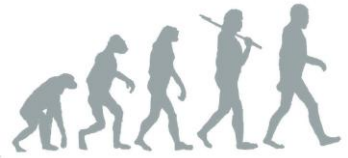
Las metaheurísticas aportan *técnicas de resolución*

APLICACIONES

- Recuperación de la arquitectura desde código fuente
- Refactorización e inclusión de patrones arquitectónicos

Introducción

Computación Evolutiva (EC)



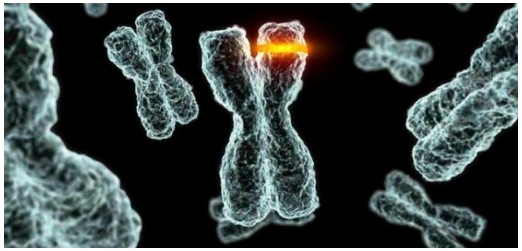
Inspirada en la evolución natural

Manejo de varias soluciones

Operadores de selección, cruce y mutación

Función de *fitness*

Programación Evolutiva (EP)



Ausencia de operador de cruce

Selección determinista

Representación adaptada al problema

Problemática

Definición de Szyperski

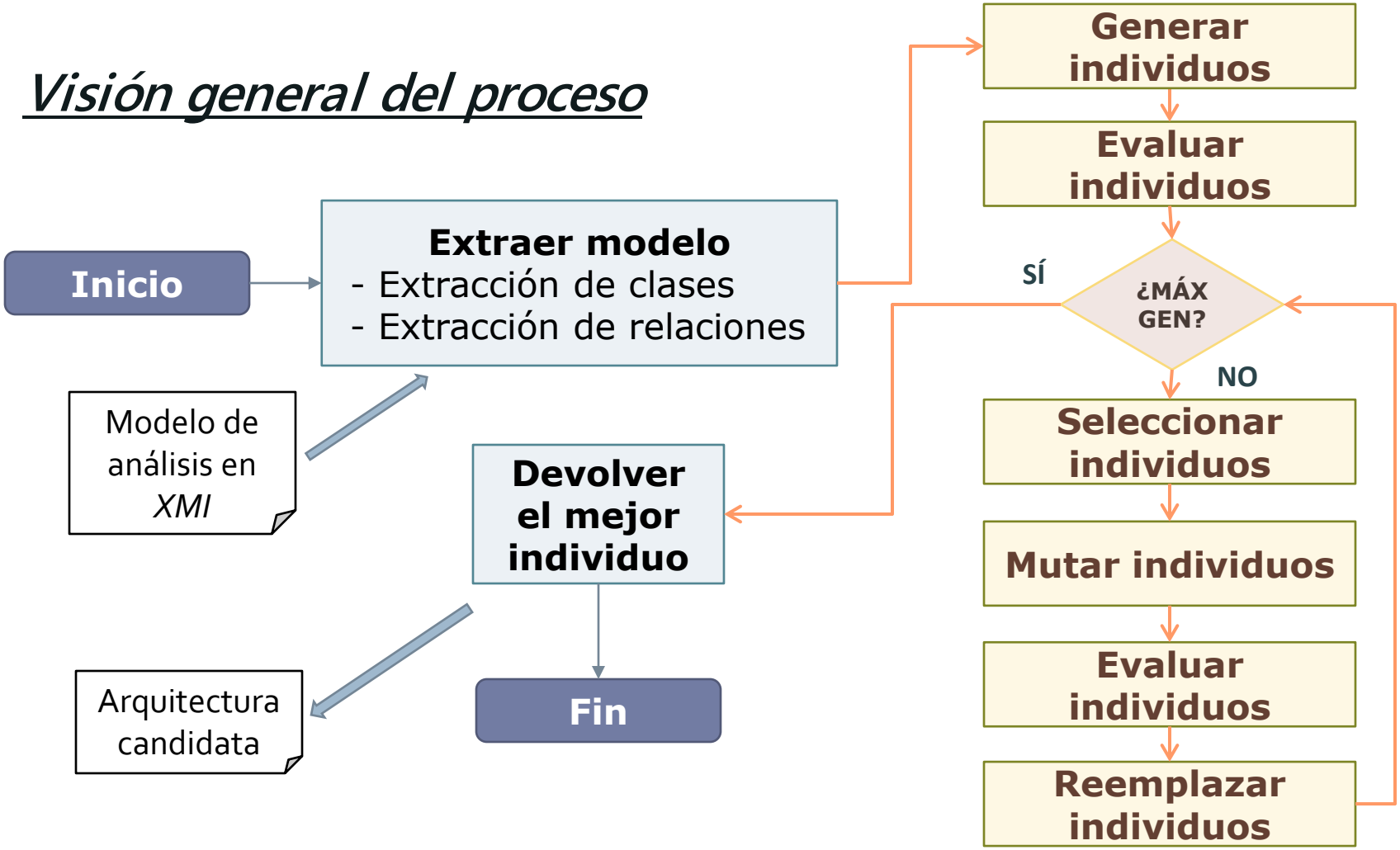
- Componente
- Interfaz
- Conector

Abstracción propuesta

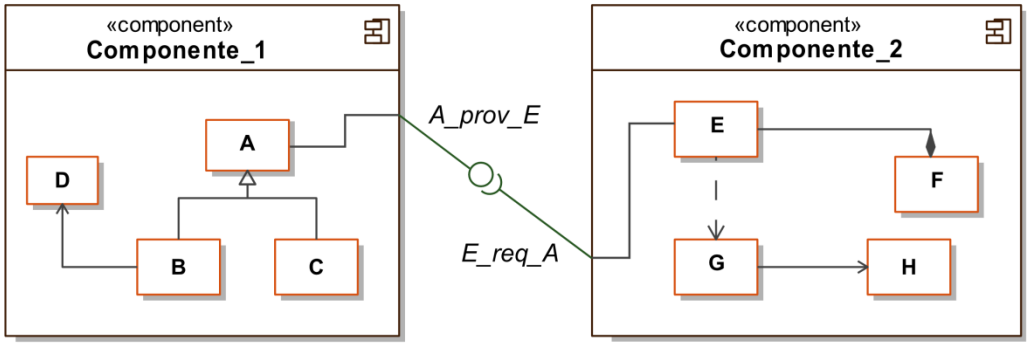
- Grupo cohesionado de clases
- Relación navegable entre clases de distintos componentes
- Unión de una pareja de interfaces, una requerida y una proveída

Diseño del algoritmo

Visión general del proceso

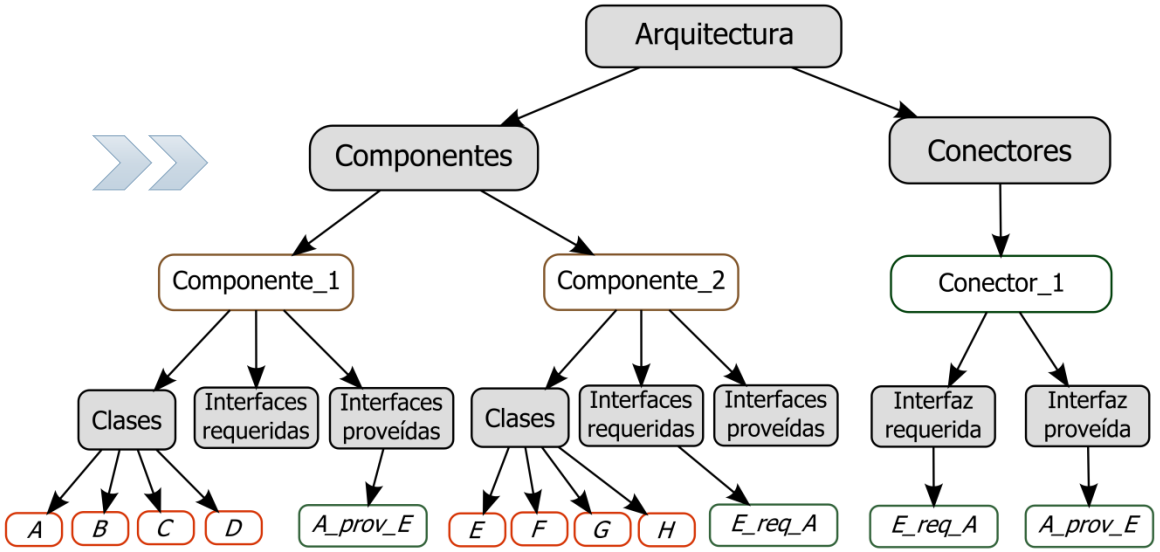


Diseño del algoritmo



Fenotipo

Genotipo



Representación del problema

Diseño del algoritmo

- Métricas asociadas a conceptos de interés para el ingeniero
  - Cohesión: basada en la "fuerza" de las intra-relaciones
  - Acoplamiento: contabiliza las inter-relaciones
  - Complejidad: asociada al tamaño de los componentes

$$coh_i = \frac{w_{c1}}{(w_{as} + w_{de} + w_{co} + w_{ge}) \cdot (n_{cl} - 1)} \cdot [w_{as} \cdot n_{as} + w_{de} \cdot n_{de} + w_{co} \cdot (n_{ag} + n_{co}) + w_{ge} + n_{ge}] + w_{c2} \cdot \frac{1 - n_{gr}}{n_{cl} - 1}$$

$$acopl = \frac{1}{R_{\max}} \cdot \frac{2 \cdot R}{C \cdot (C-1)}$$

$$R = \sum_{i=1}^{C} \sum_{j=1}^{C} \max r_{i,j}^{\ k}$$

$$CV = \frac{\sigma}{\mu}$$

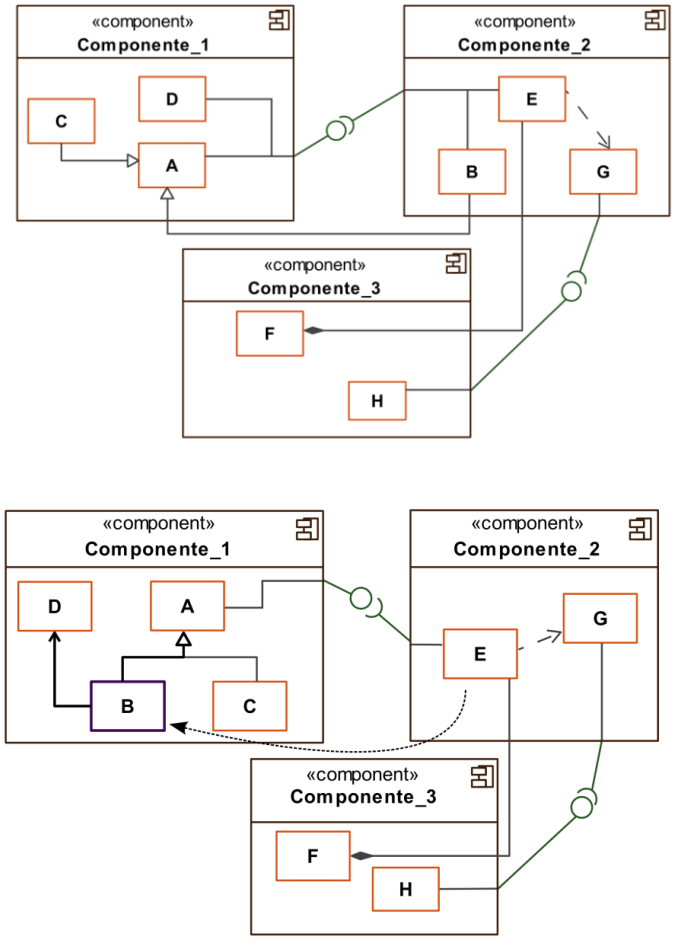
Evaluación de soluciones

$$fitness = w_{coh} \cdot media(coh) + w_{acopl} \cdot norm(inv(acopl)) + w_{cv} \cdot norm(inv(cv))$$

Diseño del algoritmo

Operadores de mutación

- Realizan pequeñas alteraciones al individuo
- Representan transformaciones arquitectónicas
  - Mover una clase
  - Añadir un componente
  - Eliminar un componente
  - Dividir un componente
  - Unir dos componentes



Diseño del algoritmo

- Inicialización
  - Distribución aleatoria de clases
  - Identificación de interfaces candidatas
  - Número de componentes aleatorio (min-máx)
- Selección
  - Cada individuo actúa como padre
- Reemplazo
  - Sobrevive el mejor entre cada padre y su descendiente
- Restricciones
  - Componentes vacíos o aislados
  - Clases replicadas
  - Componentes mutuamente dependientes

Parámetros y casos de estudio

Parámetro	Valor
T. Población	100
Nº Gener.	100
Máx. Nº. Mut.	10
W_{coh}	0,300
W_{aco}	0,400
W_{cv}	0,300

Parámetro	Valor
Prob. Mut. Añadir Comp.	0,250
Prob. Mut. Eliminar Comp.	0,125
Prob. Mut. Dividir Comp.	0,125
Prob. Mut. Unir Comp.	0,250
Prob. Mut. Mover Clase	0,250
Número de componentes	2-8

JCLEC

dataprо4j

SDMetrics®

UNIFIED MODELING LANGUAGE

magicdraw

Problema						
NekoHTML	24	16	5	0	0	10
AquaLush	58	69	6	0	0	20
Datapro4j	59	4	18	1	4	50

Resultados

- Mejores resultados en *fitness*, cohesión y acoplamiento
- Diferencia en Datapro4j (naturaleza del diseño)
- Agrupaciones originales de clases
- Mantenimiento de diversidad de arquitecturas
- Diferencias significativas (Wilcoxon Test, 90%)

Problema	Programación Evolutiva				Búsqueda Aleatoria			
	Fitness	***Coh.***	***Acopl***	***CV***	***Fitness***	***Coh.***	***Acopl.***	***CV***
NekoHTML	**0,7301**	**0,1977**	**0,0000**	0,7806	0,6198	-0,0390	0,1420	**0,3129**
AquaLush	**0,6793**	**0,1262**	**0,0600**	0,9222	0,4640	-0,2510	0,3694	**0,3460**
Datapro4j	**0,4780**	**0,0767**	**0,5200**	0,9868	0,3316	0,0541	0,9522	**0,1008**

Conclusiones y trabajo futuro

- Conclusiones
  - Modelo de identificación y optimización de componentes basado en Programación Evolutiva
  - Desarrollo de una representación y operadores específicos
  - Conceptos de diseño como función de *fitness*

- Trabajo futuro
  - Extender la información del modelo de análisis
  - Implicación del experto: interactividad
  - Aplicar otras técnicas metaheurísticas
  - Abordar otros problemas del análisis arquitectónico

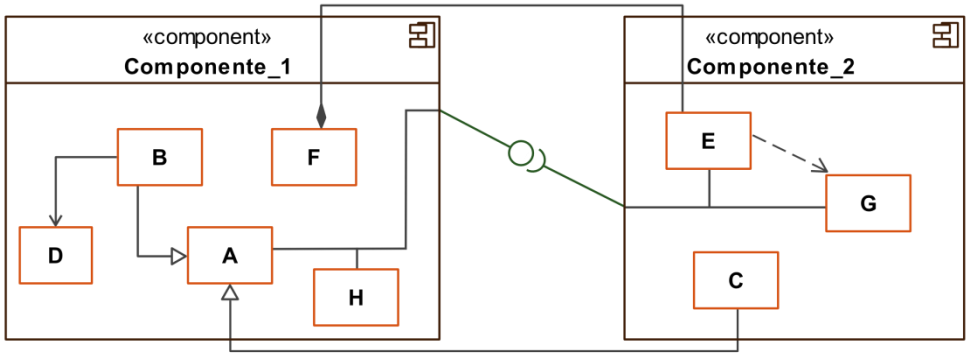
Identificación de Componentes en Arquitecturas Software Mediante Programación Evolutiva

A. Ramírez, J.R. Romero, S. Ventura

Dpto. de Informática y Análisis Numérico. Universidad de Córdoba, España.

XVIII Jornadas en Ingeniería del Software y Bases de Datos (JISBD 2013).
Madrid (España). 17-20 de septiembre de 2013.

Un ejemplo...



Cohesión

$$Componente1 \rightarrow n_{gr_1} = 3,\ n_{cl1} = 5,\ n_{as1} = 1,\ n_{ge_1} = 1$$

$$coh_1 = -0{,}1750$$

$$Componente2 \rightarrow n_{gr_2} = 2,\ n_{cl2} = 3,\ n_{de2} = 1$$

$$coh_2 = -0{,}2250$$

$$coh = \frac{-0{,}1750 + (-0{,}2250)}{2} = -0{,}2000$$

$$w_{as} = 1, \quad w_{de} = 1$$

$$w_{co} = 3 \quad w_{ge} = 5$$

$$w_{c1} = 0{,}5 \quad w_{c2} = 0{,}5$$

CV

$$\mu = \frac{5+3}{2} = 4$$

$$\sigma = \sqrt{(5-4)^2 + (3-4)^2} = 1{,}4142$$

$$CV = \frac{1{,}4142}{4} = 0{,}3536$$

Acoplamiento

$$Inter-relaciones \rightarrow r_{1,2}{}^{1} = w_{co} = 3,\ r_{1,2}{}^{2} = w_{ge} = 1$$

$$R = \max(3,5) = 5$$

$$R_{\max} = w_{ge} = 5$$

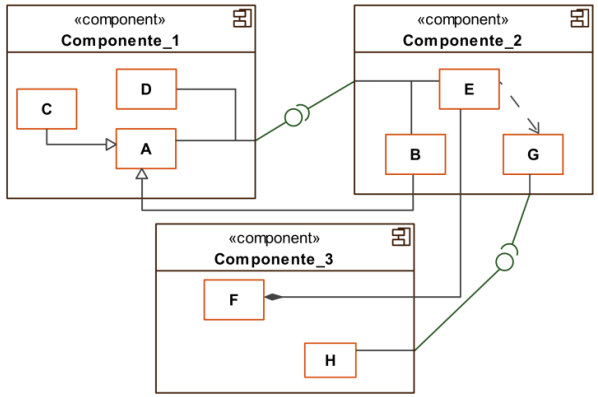
$$acopl = \frac{1}{5} \cdot \frac{2 \cdot 5 \cdot 1}{2} = 1$$

Fitness

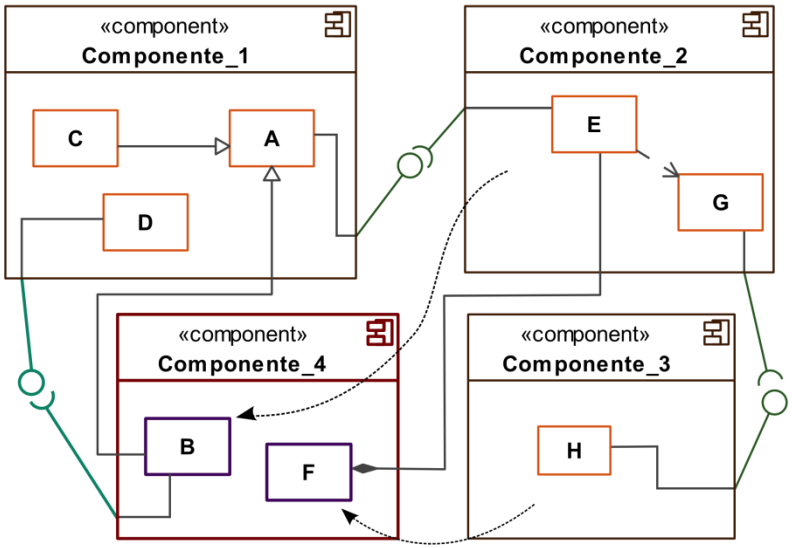
$$fitness = 0{,}3 \cdot (-0{,}2) + 0{,}4 \cdot (1-1) + 0{,}3 \cdot \left(\frac{8 - 0{,}3536}{8} \right) = 0{,}2267$$

Un ejemplo…

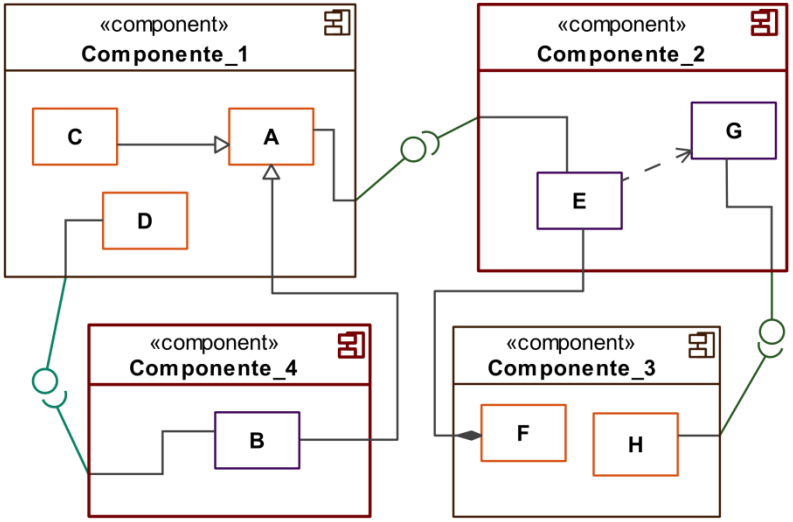
Individuo inicial



Añadir un componente

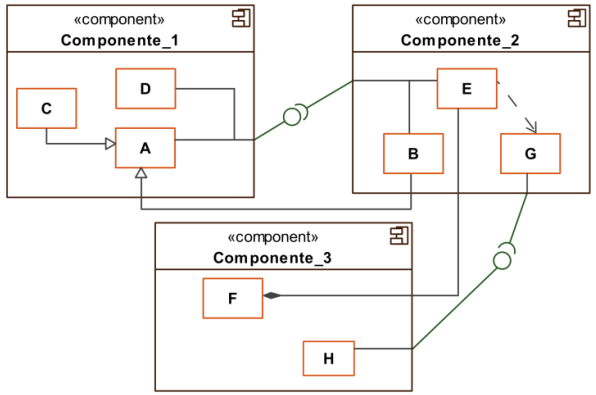


Dividir un componente

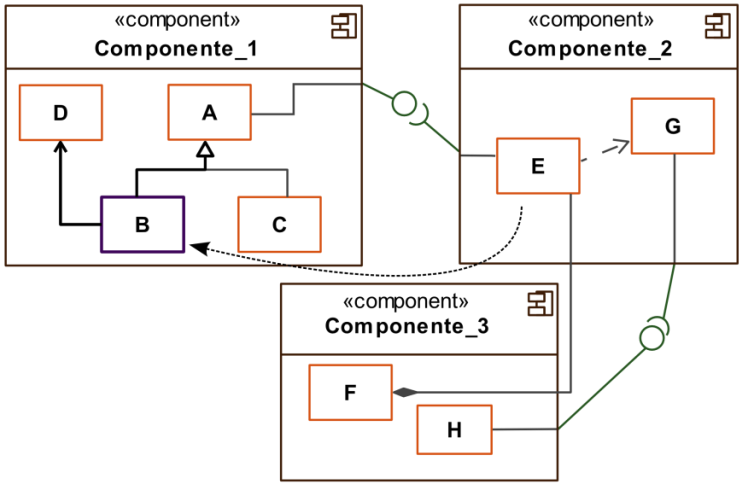


Un ejemplo…

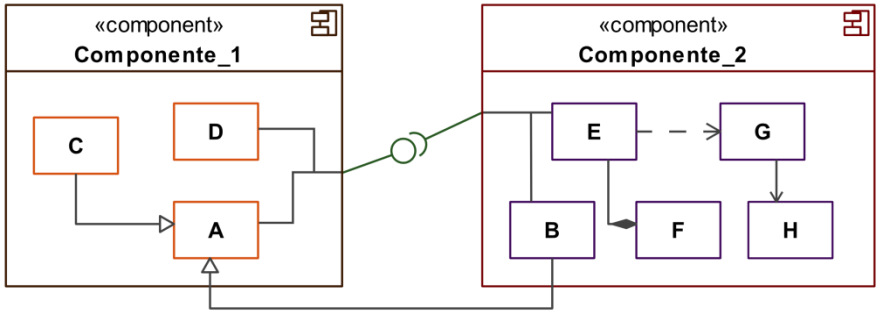
Individuo inicial



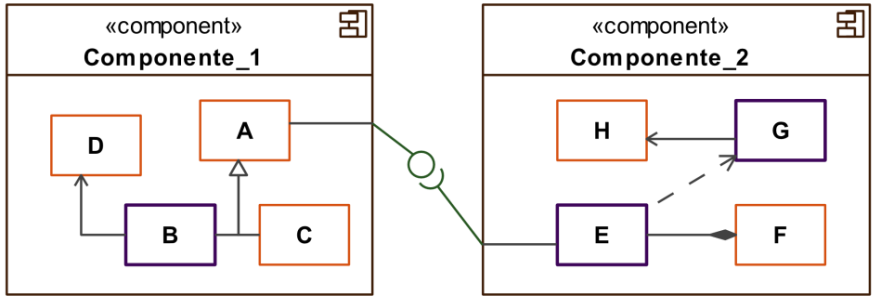
Mover una clase



Unir dos componentes

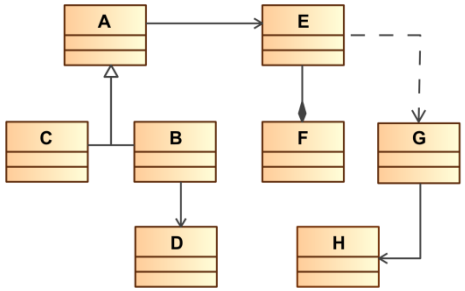


Eliminar un componente

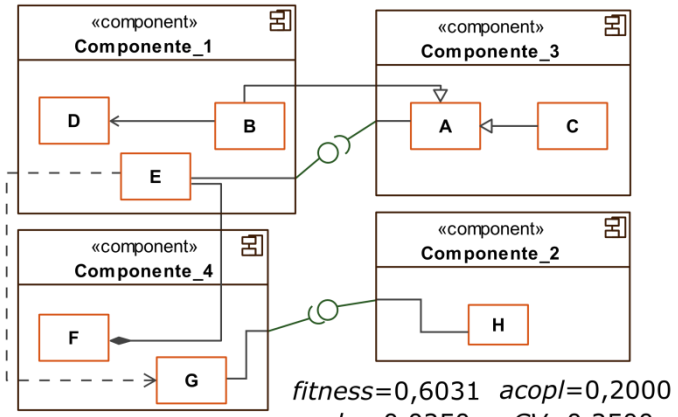


Un ejemplo…

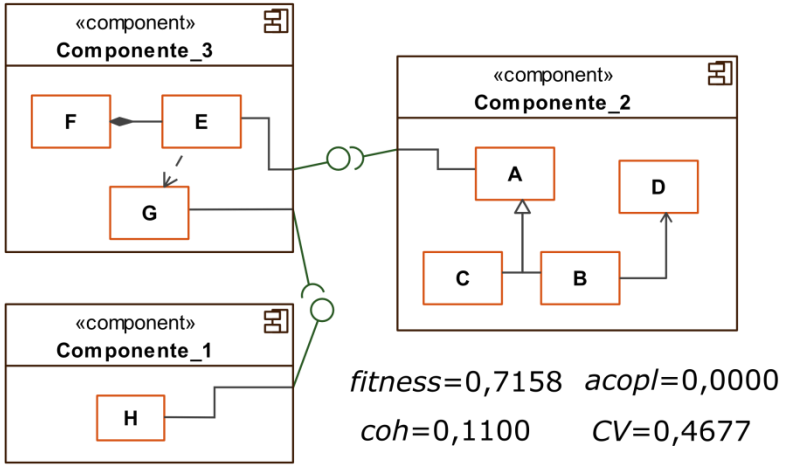
Modelo de análisis



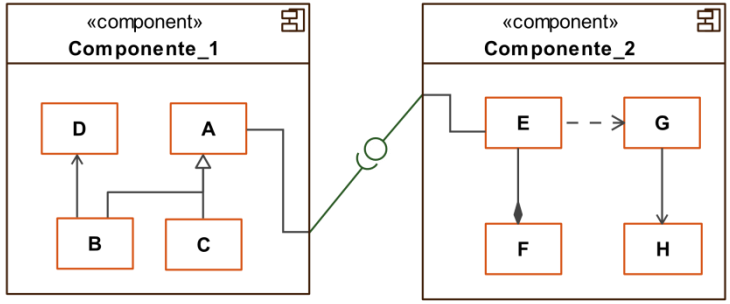
Individuo inicial



fitness=0,6031 *acopl*=0,2000
coh=-0,0250 *CV*=0,2500



fitness=0,7158 *acopl*=0,0000
coh=0,1100 *CV*=0,4677



fitness=0,7400
coh=0,1300

acopl=0,0000
CV=0,0000

Tras 5 generaciones

Tras 10 generaciones