UNIVERSIDAD
D
CÓRDOBA

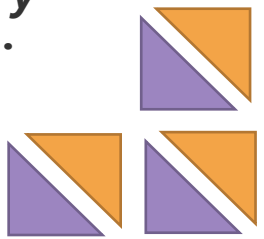
KDIS Lab
KNOWLEDGE DISCOVERY AND INTELLIGENT SYSTEMS

Algoritmo de programación evolutiva para identificación de arquitecturas software

A. Ramírez, J.R. Romero, S. Ventura

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

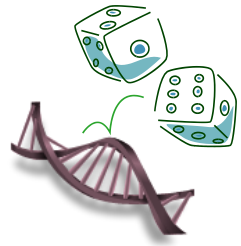
***IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*. Madrid (España). 17-20 de septiembre de 2013.**



Índice de contenido

Introducción

- Ingeniería del Software
  - Tareas muy vinculadas a decisiones humanas
  - Repercusión sobre la calidad del software
- Metaheurísticas Bio-inspiradas
  - Resolución de problemas de optimización y búsqueda
  - Utilizadas con éxito en múltiples campos



¿Qué es *S*earch-*B*ased *S*oftware *E*ngineering?

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

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

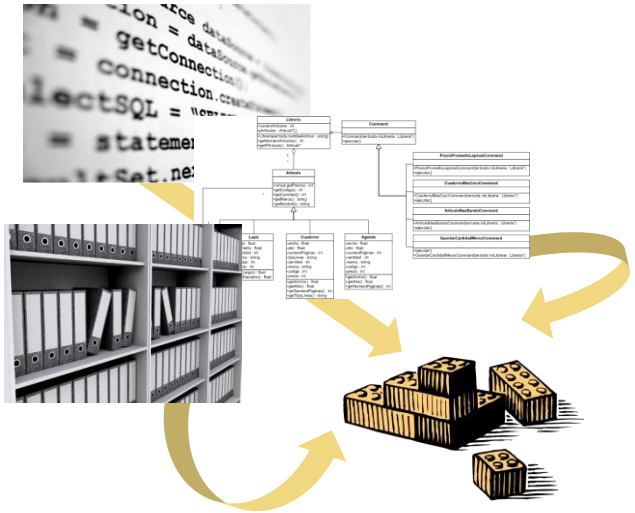
Introducción

Análisis arquitectónico

Definir los bloques funcionales del software

Útil en sistemas complejos

Calidad, reusabilidad y mantenimiento



Recuperación de la arquitectura

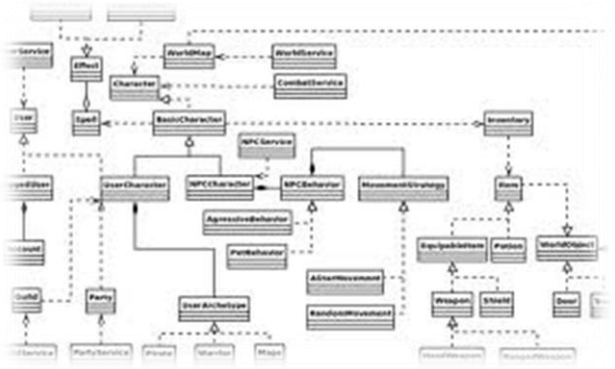
Abstraer la funcionalidad de sistemas ya operativos

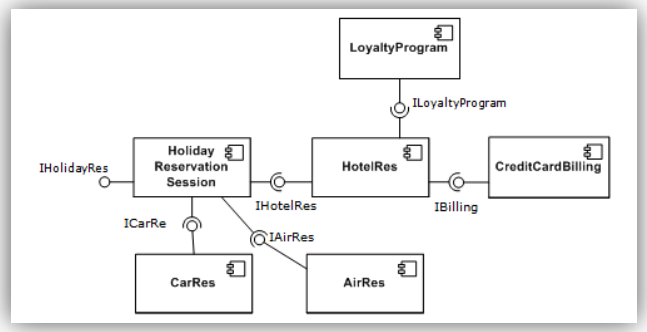
A partir de modelos de análisis cercanos a la implementación

Enfocado como problema de optimización

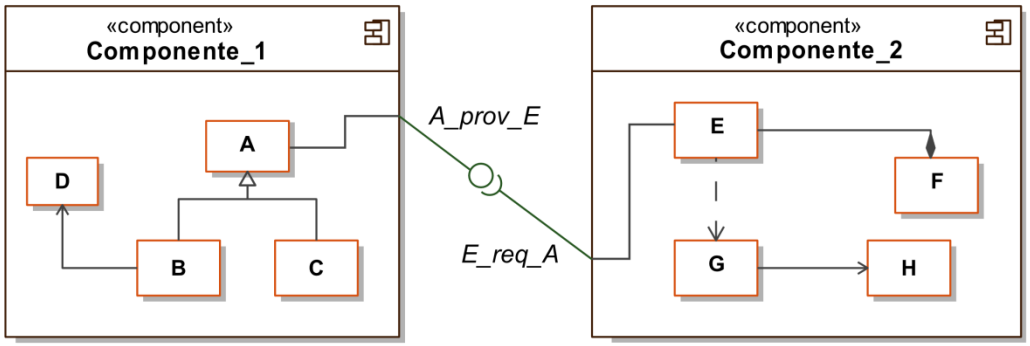
El problema

- Método de apoyo a la decisión del arquitecto software
- Búsqueda de la configuración óptima de:
  - Componentes, Interfaces y Conectores
- Partiendo de un modelo de análisis detallado del sistema
  - Búsqueda de agrupaciones de clases fuertemente relacionadas
- Presencia de restricciones



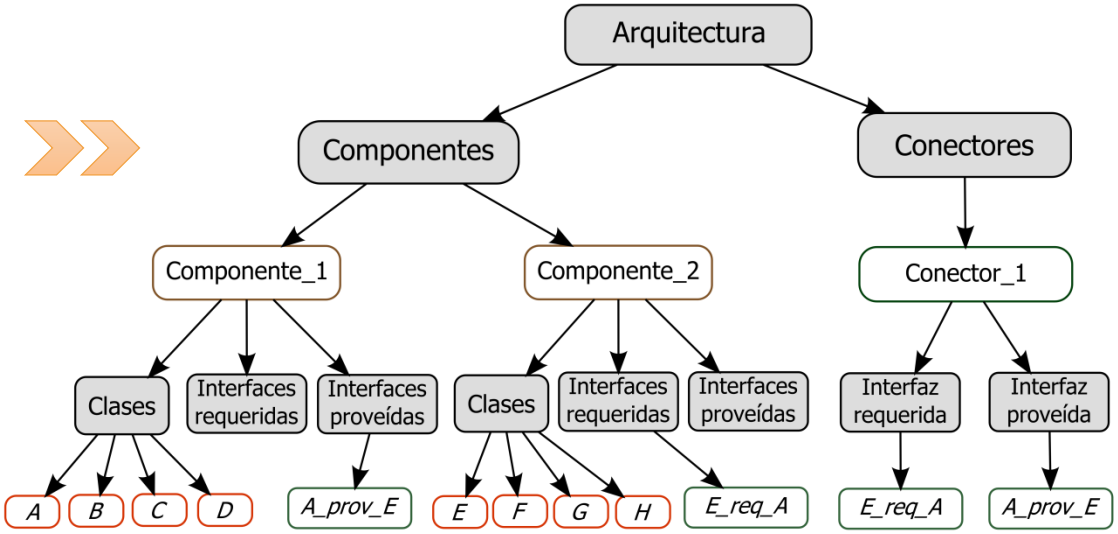


Diseño del algoritmo
Genotipo y fenotipo



Fenotipo

Genotipo



Flexible
Eficiente
Interpretable
Similitud con modelos

Diseño del algoritmo
Función de *fitness*

- Métricas asociadas a conceptos de diseño
  - Cohesión, relacionada con las intra-relaciones
  - Acoplamiento, relacionada con las inter-relaciones
  - Complejidad, relacionada con el tamaño

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

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

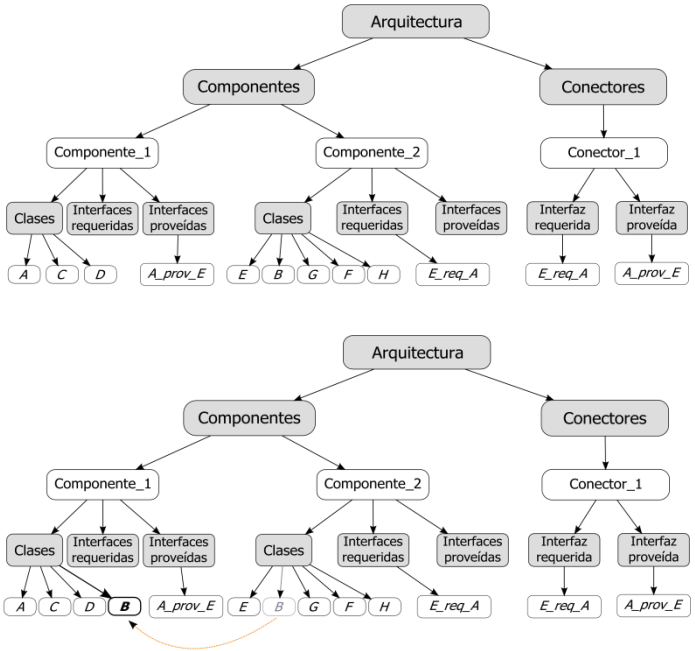
$$R = \sum_{i=1}^{C} \sum_{j=1}^{C} \max r_{i,j}^{\,k}$$

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

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

Diseño del algoritmo
Operadores genéticos

- Ausencia de operador de cruce
- Cinco operadores de mutación
  - Representan transformaciones arquitectónicas
    - *Mover una clase*
    - *Añadir un componente*
    - *Eliminar un componente*
    - *Unir dos componentes*
    - *Dividir un componente*



- Selección del operador basada en una ruleta de probabilidades

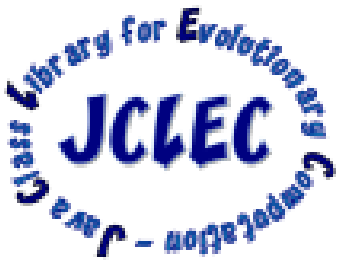
Diseño del algoritmo
Inicialización, selección y reemplazo

- Inicialización
  - Número de componentes aleatorio (min-máx.)
  - Distribución aleatoria de las clases
  - Identificación de interfaces candidatas
- Selección
  - Cada individuo actúa como padre
- Reemplazo
  - Sobrevive el mejor entre el padre y su descendiente
  - Mejor estrategia entre las consideradas

Experimentación
Configuración y parámetros

Parámetro	Valor
T. Población	100
Nº Generaciones	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



datapro4j

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

SDMetrics®

Experimentación
Resultados obtenidos

- Introducción
  - Mejores valores en media para *fitness*, cohesión y acoplamiento
  - Encuentra agrupaciones originales de clases
  - Mantiene la diversidad de arquitecturas
  - Diferencias significativas (Wilcoxon, 90%)

Problema	Programación Evolutiva				Búsqueda Aleatoria			
	Fitness	*Coh.*	*Acopl.*	*CV*	*Fitness*	*Coh.*	*Acopl.*	*CV*
NekoHTML	**0,7301 ± 0,0014**	**0,1977 ± 0,0105**	**0,0000 ± 0,0000**	0,7806 ± 0,1208	0,6198 ± 0,0402	-0,0390 ± 0,0828	0,1420 ± 0,0776	**0,3129 ± 0,2576**
AquaLush	**0,6793 ± 0,0254**	**0,1262 ± 0,0446**	**0,0600 ± 0,0899**	0,9222 ± 0,0754	0,4640 ± 0,0112	-0,2510 ± 0,0401	0,3694 ± 0,0346	**0,3460 ± 0,1014**
Datapro4j	**0,4780 ± 0,0206**	**0,0767 ± 0,0991**	**0,5200 ± 0,0623**	0,9868 ± 0,2571	0,3316 ± 0,0102	0,0541 ± 0,0980	0,9522 ± 0,1001	**0,1008 ± 0,0963**

Conclusiones y trabajo futuro

Conclusiones

Algoritmo para la optimización de arquitecturas software

Novedad en el enfoque del problema

Representación y operadores específicos

Fitness inspirado en conceptos habituales en I. Software

Trabajo futuro

Creación de nuevos operadores

Aplicación de otras metaheurísticas o su hibridación

Inclusión de interactividad con el experto

UNIVERSIDAD
D
CÓRDOBA

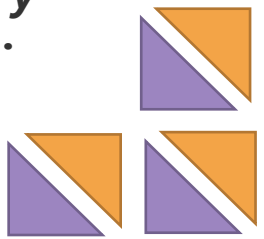
KDIS Lab
KNOWLEDGE DISCOVERY AND INTELLIGENT SYSTEMS

Algoritmo de programación evolutiva para identificación de arquitecturas software

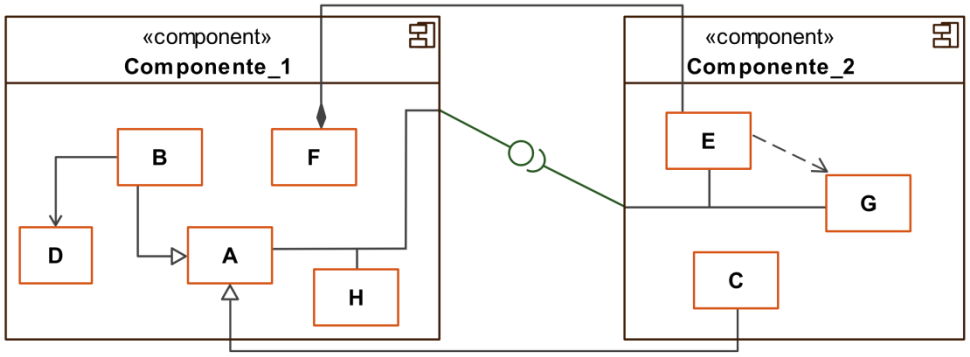
A. Ramírez, J.R. Romero, S. Ventura

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

***IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*. Madrid (España). 17-20 de septiembre de 2013.**



Un ejemplo...



Cohesión

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

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

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

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

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

Acoplamiento

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

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

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

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

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

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

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

CV

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

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

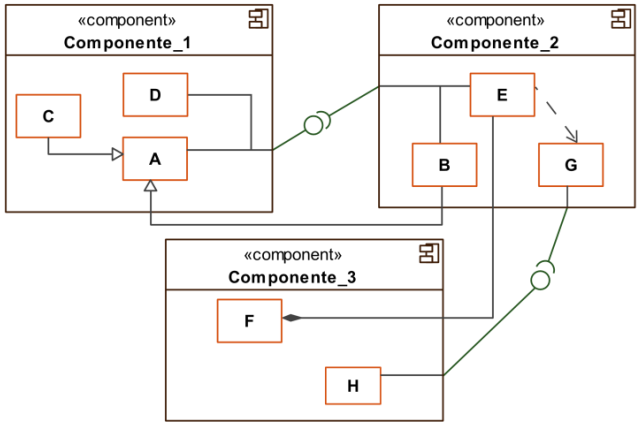
$$CV = \frac{1{,}4142}{4} = 0{,}3536$$

Fitness

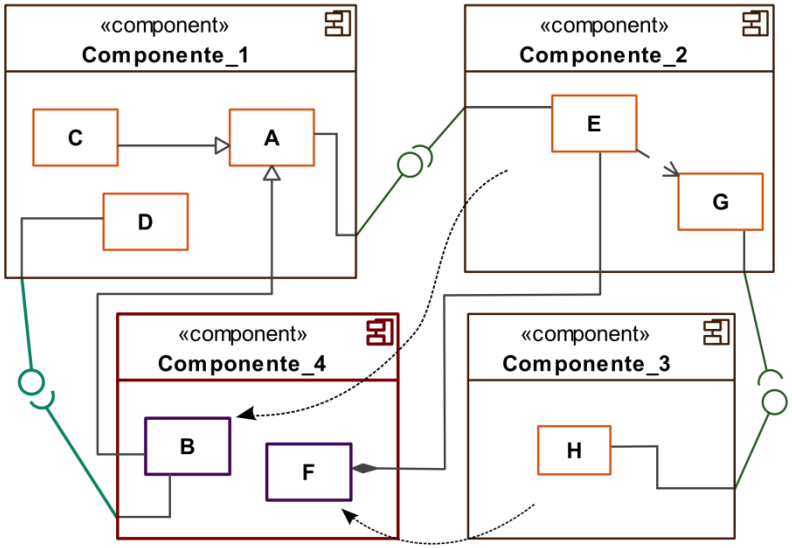
$$fitness = 0{,}2267$$

Un ejemplo...

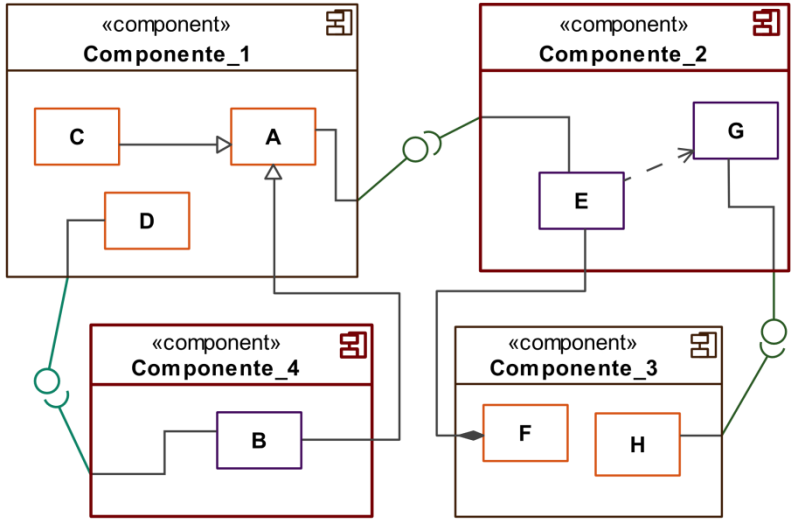
Individuo inicial



Añadir un componente

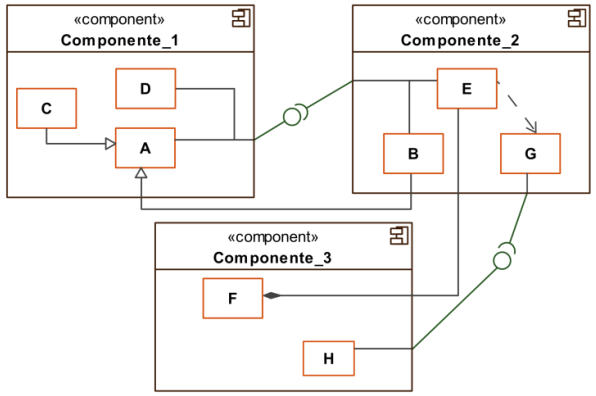


Dividir un componente

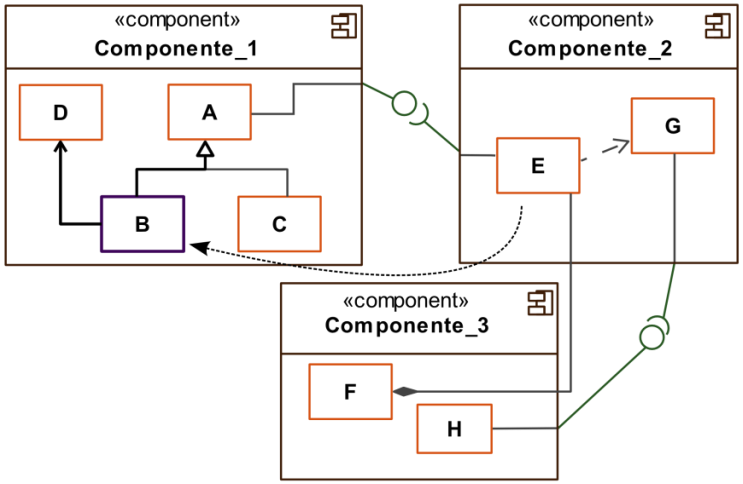


Un ejemplo...

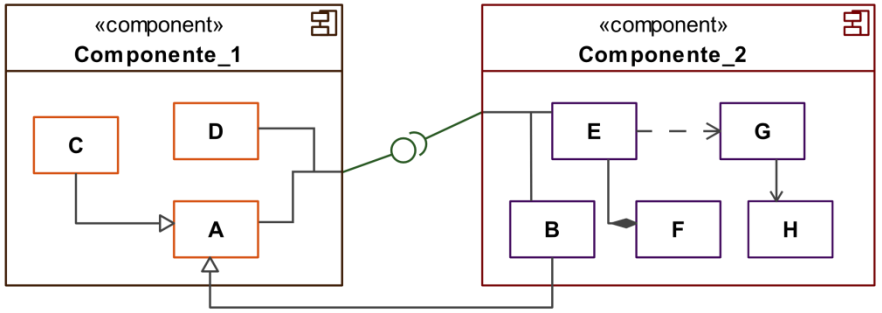
Individuo inicial



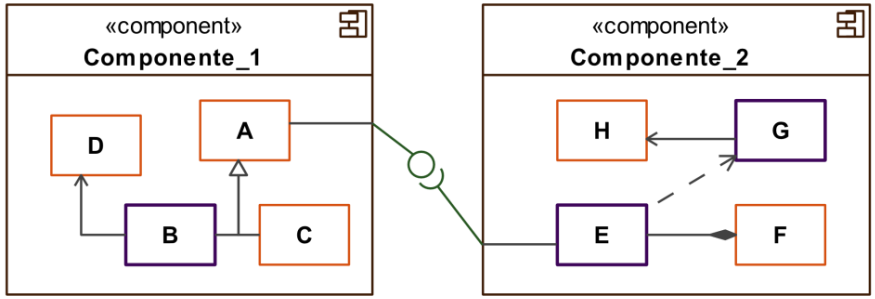
Mover una clase



Unir dos componentes

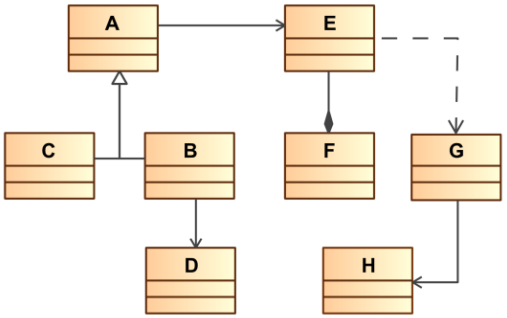


Eliminar un componente

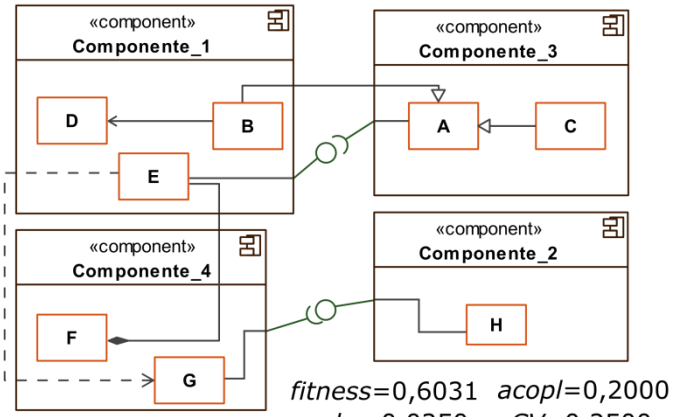


Un ejemplo...

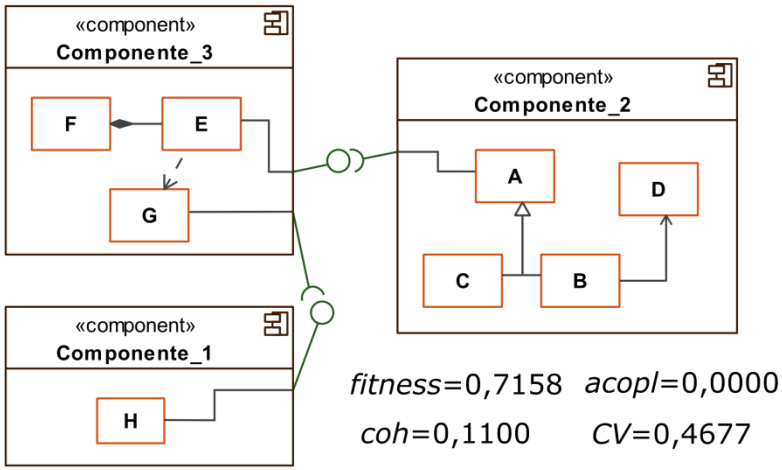
Modelo de análisis



Individuo inicial

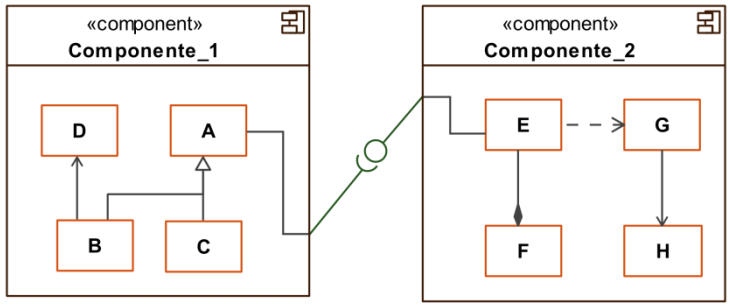


fitness=0,6031 *acopl*=0,2000
coh=-0,0250 *CV*=0,2500



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

Tras 5 generaciones



fitness=0,7400
coh=0,1300

acopl=0,0000
CV=0,0000

Tras 10 generaciones