

Change-Oriented Software Engineering

Peter Ebraert, Jorge Vallejos, Pascal Costanza, Ellen Van Paesschen and Theo D'Hondt



Vrije Universiteit Brussel

Change-oriented SW Eng.

- Software Development
 - ≠ Writing a text
 - Constructing a system by means of building blocks

Change-oriented SW Eng.

- Software Development
 - ≠ Writing a text
 - Constructing a system by means of building blocks
- Construction
 - Changing a construction yard towards a required structure

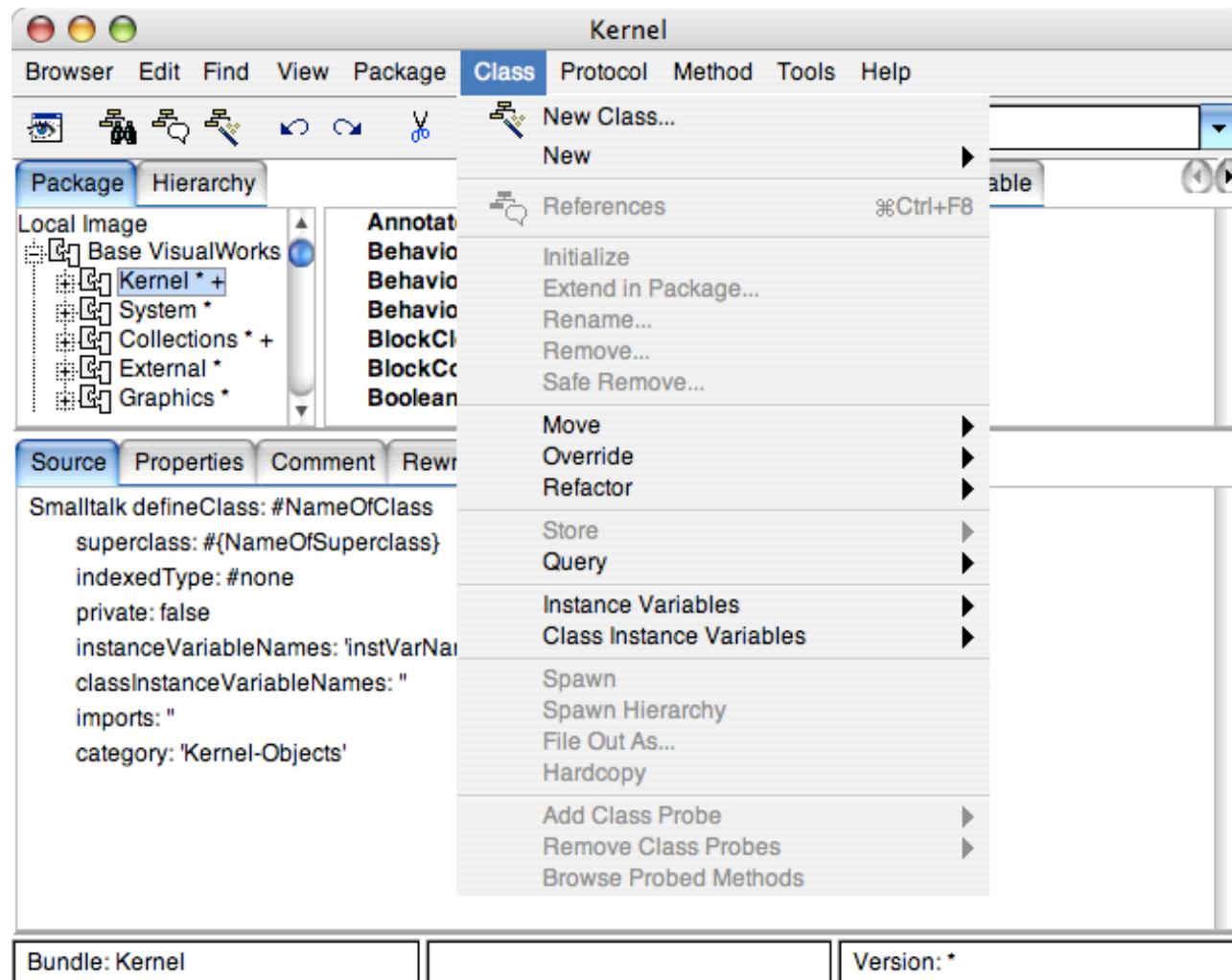
Change-oriented SW Eng.

- Software Development
 - ≠ Writing a text
 - Constructing a system by means of building blocks
- Construction
 - Changing a construction yard towards a required structure
- Software Development

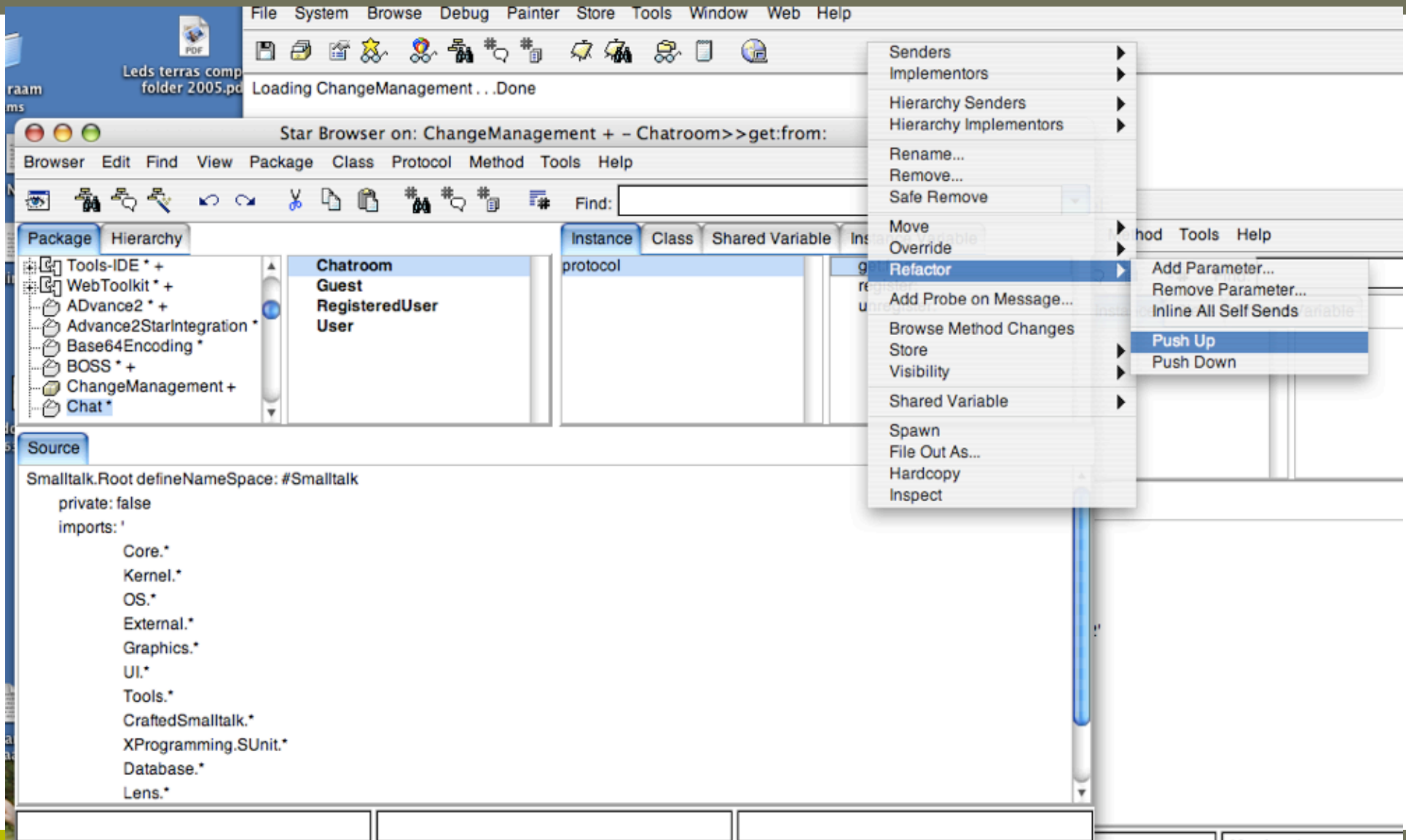
Change-oriented SW Eng.

- Software Development
 - \neq Writing a text
 - Constructing a system by means of building blocks
- Construction
 - Changing a construction yard towards a required structure
- Software Development \in Software evolution

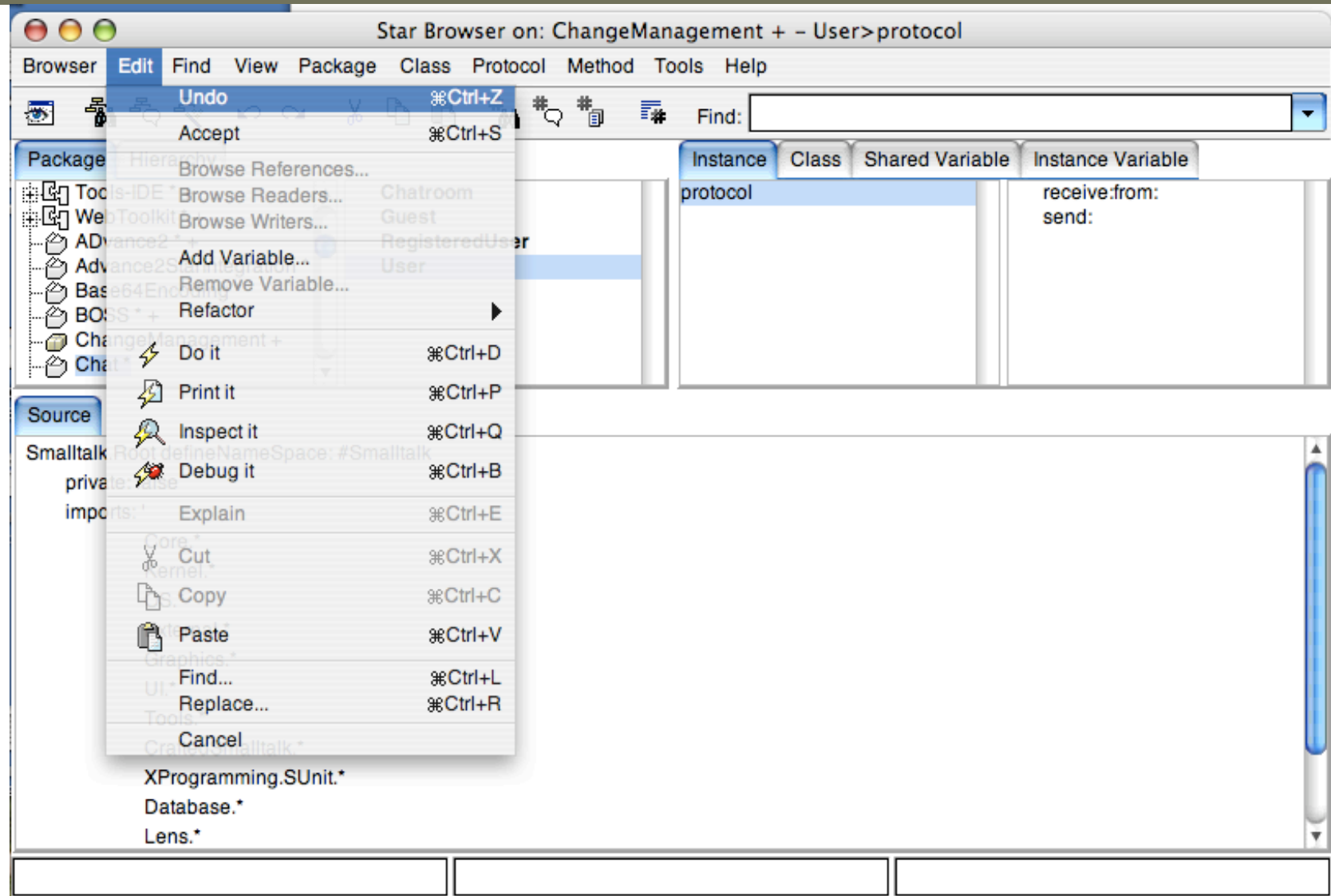
Change-Oriented Support



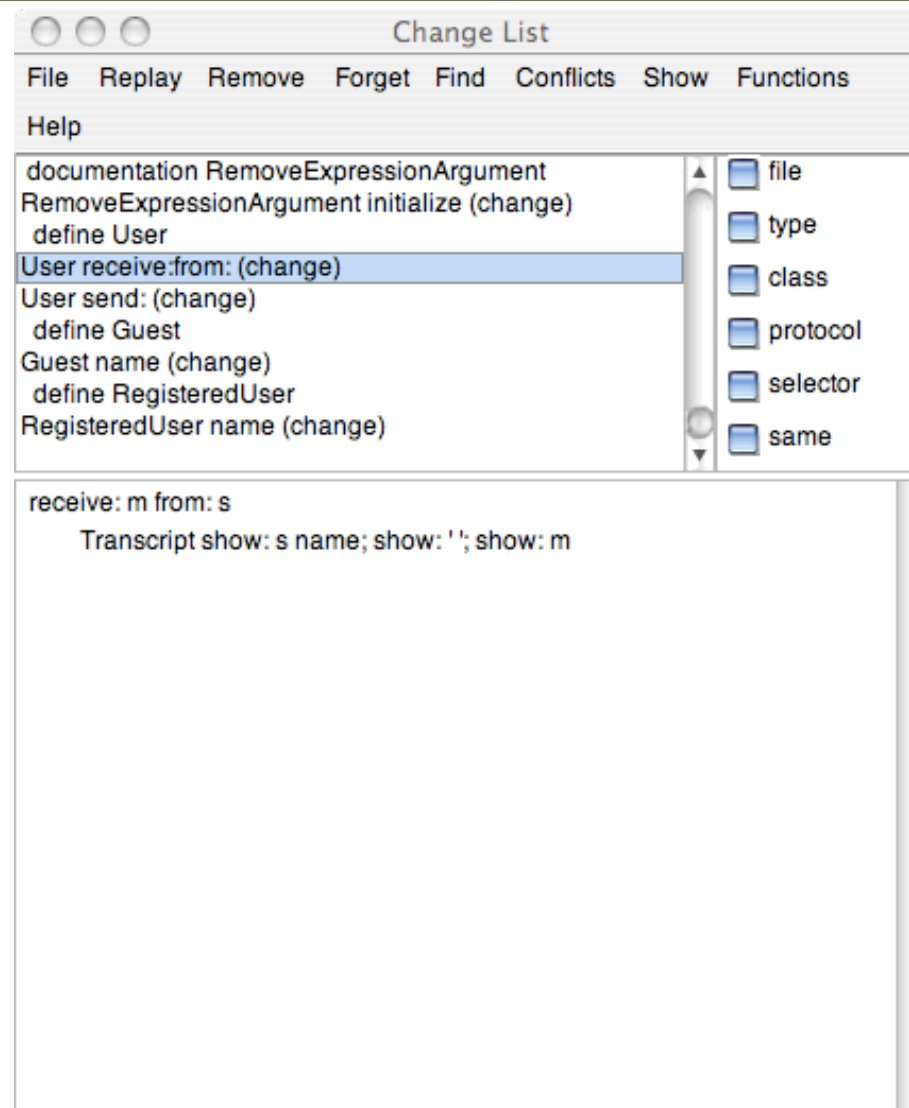
Change-Oriented Support



Change-Oriented Support?



Change-Oriented Support?



What do we want?

- Support for changes in forward engineering
- Undo mechanism
- Readable change list
- Change objects which can be reused
- Changes should be applicable on different software systems / different platforms

What do we want?

- Support for changes in forward engineering
- Undo mechanism
 - ▶ **Evolvability & Maintainability**
- Readable change list
- Change objects which can be reused
- Changes should be applicable on different software systems / different platforms

What do we want?

- Support for changes in forward engineering
- Undo mechanism
 - **Evolvability & Maintainability**
- Readable change list
 - **Understandability**
- Change objects which can be reused
- Changes should be applicable on different software systems / different platforms

What do we want?

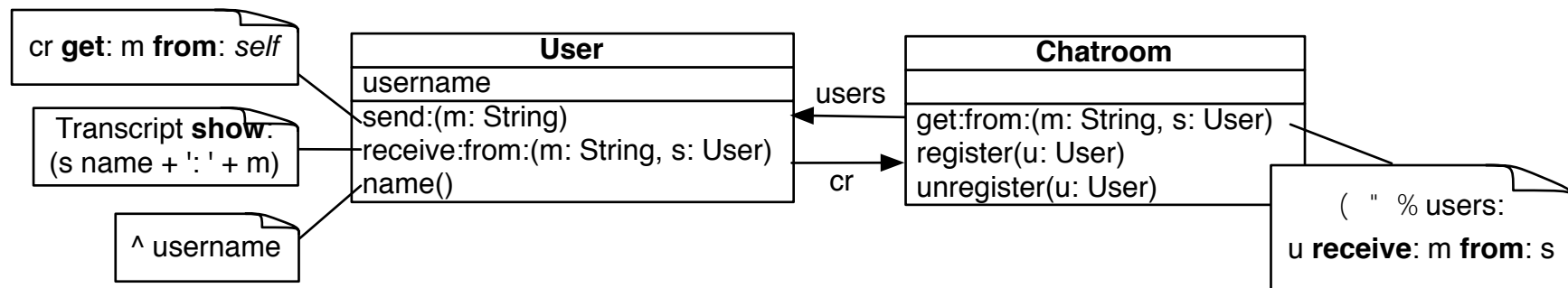
- Support for changes in forward engineering
- Undo mechanism
 - ▶ **Evolvability & Maintainability**
- Readable change list
 - ▶ **Understandability**
- Change objects which can be reused
- Changes should be applicable on different software systems / different platforms
 - ▶ **Reusability**

Is there a solution around?

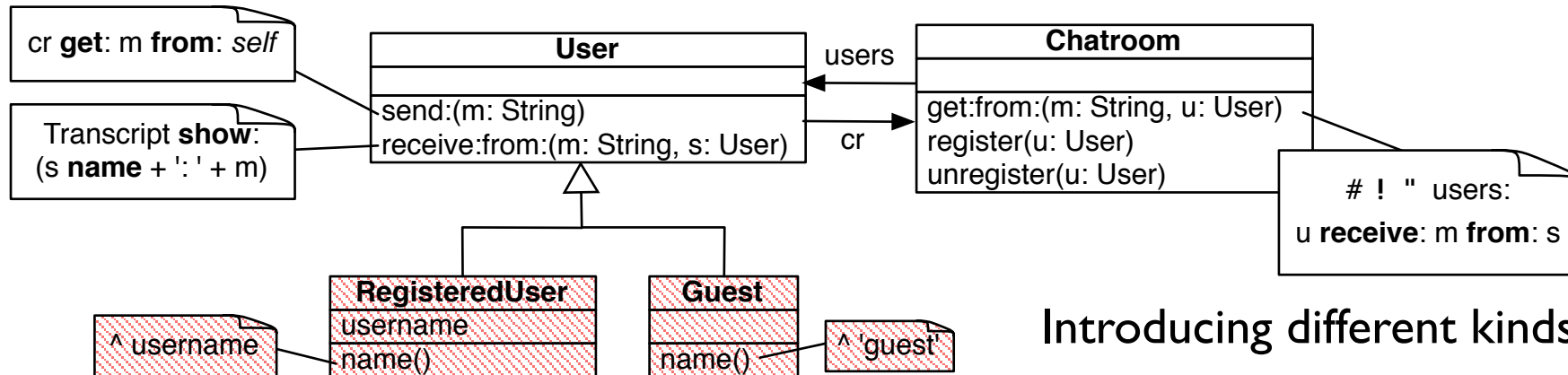
- Centralise Change in the development process
- First-class change objects
 - A Model for first-class changes
- What models exist already?
 - **Spyware** change model
 - **ChangeList** change model

Example case

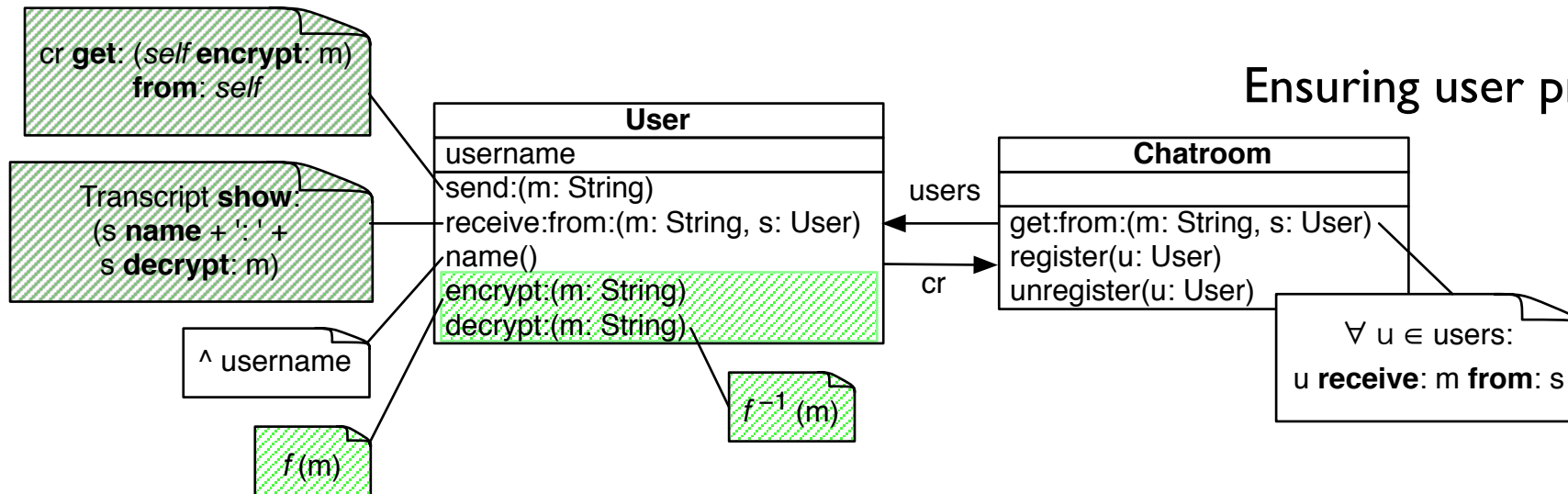
Example case



Evolution scenario



Introducing different kinds of users



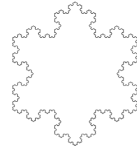
Ensuring user privacy

ChangeList's Change Log

```
1 Created package ChatApp
2 define User
3 dolt User organisation addCategory:#messaging
4 User receive:from: (change)
5 define User
6 define User
7 User send: (change)
8 dolt User organisation addCategory:#accessing
9 User name (change)
10 define Chatroom
11 define Chatroom
12 dolt Chatroom organisation addCategory:#messaging
13 Chatroom get:from: (change)
14 doit Chatroom organisation addCataegory:#registering
15 Chatroom register: (change)
16 Chatroom unregister: (change)
17
18 define Guest
19 dolt Guest organisation addCategory:#messaging
20 Guest name (change)
21 define RegisteredUser
22 RegisteredUser name (change)
23 User name (remove)
24 define User
25 define RegisteredUser
26
27 dolt User organisation addCategory:#encryption
28 User encrypt: (change)
29 User decrypt: (change)
30 User send: (change)
31 User receive:from: (change)
```

ChangeList's Change Log

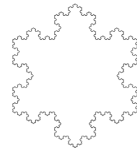
- 1 Created package ChatApp
- 2 define User
- 3 dolt User organisation addCategory:#messaging
- 4 User receive:from: (change)
- 5 define User
- 6 define User
- 7 User send: (change)
- 8 dolt User organisation addCategory:#accessing
- 9 User name (change)
- 10 define Chatroom
- 11 define Chatroom
- 12 dolt Chatroom organisation addCategory:#messaging
- 13 Chatroom get:from: (change)
- 14 doit Chatroom organisation addCataegory:#registering
- 15 Chatroom register: (change)
- 16 Chatroom unregister: (change)
- 17
- 18 define Guest
- 19 dolt Guest organisation addCategory:#messaging
- 20 Guest name (change)
- 21 define RegisteredUser
- 22 RegisteredUser name (change)
- 23 User name (remove)
- 24 define User
- 25 define RegisteredUser
- 26
- 27 dolt User organisation addCategory:#encryption
- 28 User encrypt: (change)
- 29 User decrypt: (change)
- 30 User send: (change)
- 31 User receive:from: (change)



Restricted Level Of Granularity

ChangeList's Change Log

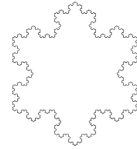
1 Created package ChatApp
2 define User
3 dolt User organisation addCategory:#messaging
4 User receive:from: (change)
5 define User
6 define User
7 User send: (change)
8 dolt User organisation addCategory:#accessing
9 User name (change)
10 define Chatroom
11 define Chatroom
12 dolt Chatroom organisation addCategory:#messaging
13 Chatroom get:from: (change)
14 doit Chatroom organisation addCataegory:#registering
15 Chatroom register: (change)
16 Chatroom unregister: (change)
17
18 define Guest
19 dolt Guest organisation addCategory:#messaging
20 Guest name (change)
21 define RegisteredUser
22 RegisteredUser name (change)
23 User name (remove)
24 define User
25 define RegisteredUser
26
27 dolt User organisation addCategory:#encryption
28 User encrypt: (change)
29 User decrypt: (change)
30 User send: (change)
31 User receive:from: (change)



Restricted Level Of Granularity

ChangeList's Change Log

```
1 Created package ChatApp
2 define User
3 dolt User organisation addCategory:#messaging
4 User receive:from: (change)
5 define User
6 define User
7 User send: (change)
8 dolt User organisation addCategory:#accessing
9 User name (change)
10 define Chatroom
11 define Chatroom
12 dolt Chatroom organisation addCategory:#messaging
13 Chatroom get:from: (change)
14 doit Chatroom organisation addCataegory:#registering
15 Chatroom register: (change)
16 Chatroom unregister: (change)
17
18 define Guest
19 dolt Guest organisation addCategory:#messaging
20 Guest name (change)
21 define RegisteredUser
22 RegisteredUser name (change)
23 User name (remove)
24 define User
25 define RegisteredUser
26
27 dolt User organisation addCategory:#encryption
28 User encrypt: (change)
29 User decrypt: (change)
30 User send: (change)
31 User receive:from: (change)
```



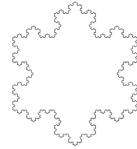
Restricted Level Of Granularity



Term Overloading

ChangeList's Change Log

```
1 Created package ChatApp
2 define User
3 dolt User organisation addCategory:#messaging
4 User receive:from: (change)
5 define User
6 define User
7 User send: (change)
8 dolt User organisation addCategory:#accessing
9 User name (change)
10 define Chatroom
11 define Chatroom
12 dolt Chatroom organisation addCategory:#messaging
13 Chatroom get:from: (change)
14 doit Chatroom organisation addCataegory:#registering
15 Chatroom register: (change)
16 Chatroom unregister: (change)
17
18 define Guest
19 dolt Guest organisation addCategory:#messaging
20 Guest name (change)
21 define RegisteredUser
22 RegisteredUser name (change)
23 User name (remove)
24 define User
25 define RegisteredUser
26
27 dolt User organisation addCategory:#encryption
28 User encrypt: (change)
29 User decrypt: (change)
30 User send: (change)
31 User receive:from: (change)
```



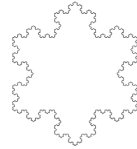
Restricted Level Of Granularity



Term Overloading

ChangeList's Change Log

```
1 Created package ChatApp
2 define User
3 dolt User organisation addCategory:#messaging
4 User receive:from: (change)
5 define User
6 define User
7 User send: (change)
8 dolt User organisation addCategory:#accessing
9 User name (change)
10 define Chatroom
11 define Chatroom
12 dolt Chatroom organisation addCategory:#messaging
13 Chatroom get:from: (change)
14 doit Chatroom organisation addCataegory:#registering
15 Chatroom register: (change)
16 Chatroom unregister: (change)
17
18 define Guest
19 dolt Guest organisation addCategory:#messaging
20 Guest name (change)
21 define RegisteredUser
22 RegisteredUser name (change)
23 User name (remove)
24 define User
25 define RegisteredUser
26
27 dolt User organisation addCategory:#encryption
28 User encrypt: (change)
29 User decrypt: (change)
30 User send: (change)
31 User receive:from: (change)
```



Restricted Level Of Granularity



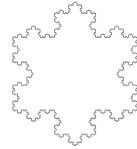
Term Overloading



Lack Of High-level Changes

ChangeList's Change Log

```
1 Created package ChatApp
2 define User
3 dolt User organisation addCategory:#messaging
4 User receive:from: (change)
5 define User
6 define User
7 User send: (change)
8 dolt User organisation addCategory:#accessing
9 User name (change)
10 define Chatroom
11 define Chatroom
12 dolt Chatroom organisation addCategory:#messaging
13 Chatroom get:from: (change)
14 doit Chatroom organisation addCataegory:#registering
15 Chatroom register: (change)
16 Chatroom unregister: (change)
17
18 define Guest
19 dolt Guest organisation addCategory:#messaging
20 Guest name (change)
21 define RegisteredUser
22 RegisteredUser name (change)
23 User name (remove)
24 define User
25 define RegisteredUser
26
27 dolt User organisation addCategory:#encryption
28 User encrypt: (change)
29 User decrypt: (change)
30 User send: (change)
31 User receive:from: (change)
```



Restricted Level Of Granularity



Term Overloading



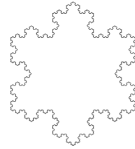
Lack Of High-level Changes

User Kinds

User Privacy

ChangeList's Change Log

```
1 Created package ChatApp
2 define User
3 dolt User organisation addCategory:#messaging
4 User receive:from: (change)
5 define User
6 define User
7 User send: (change)
8 dolt User organisation addCategory:#accessing
9 User name (change)
10 define Chatroom
11 define Chatroom
12 dolt Chatroom organisation addCategory:#messaging
13 Chatroom get:from: (change)
14 doit Chatroom organisation addCataegory:#registering
15 Chatroom register: (change)
16 Chatroom unregister: (change)
17
18 define Guest
19 dolt Guest organisation addCategory:#messaging
20 Guest name (change)
21 define RegisteredUser
22 RegisteredUser name (change)
23 User name (remove)
24 define User
25 define RegisteredUser
26
27 dolt User organisation addCategory:#encryption
28 User encrypt: (change)
29 User decrypt: (change)
30 User send: (change)
31 User receive:from: (change)
```



Restricted Level Of Granularity



Term Overloading



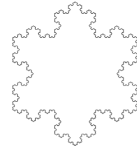
Lack Of High-level Changes



No exploration facilities

ChangeList's Change Log

```
1 Created package ChatApp
2 define User
3 dolt User organisation addCategory:#messaging
4 User receive:from: (change)
5 define User
6 define User
7 User send: (change)
8 dolt User organisation addCategory:#accessing
9 User name (change)
10 define Chatroom
11 define Chatroom
12 dolt Chatroom organisation addCategory:#messaging
13 Chatroom get:from: (change)
14 doit Chatroom organisation addCataegory:#registering
15 Chatroom register: (change)
16 Chatroom unregister: (change)
17
18 define Guest
19 dolt Guest organisation addCategory:#messaging
20 Guest name (change)
21 define RegisteredUser
22 RegisteredUser name (change)
23 User name (remove)
24 define User
25 define RegisteredUser
26
27 dolt User organisation addCategory:#encryption
28 User encrypt: (change)
29 User decrypt: (change)
30 User send: (change)
31 User receive:from: (change)
```



Restricted Level Of Granularity



Term Overloading



Lack Of High-level Changes



No exploration facilities

=> Bad for all the “ilities”

Model of First-class Changes

Fine-grained Changes

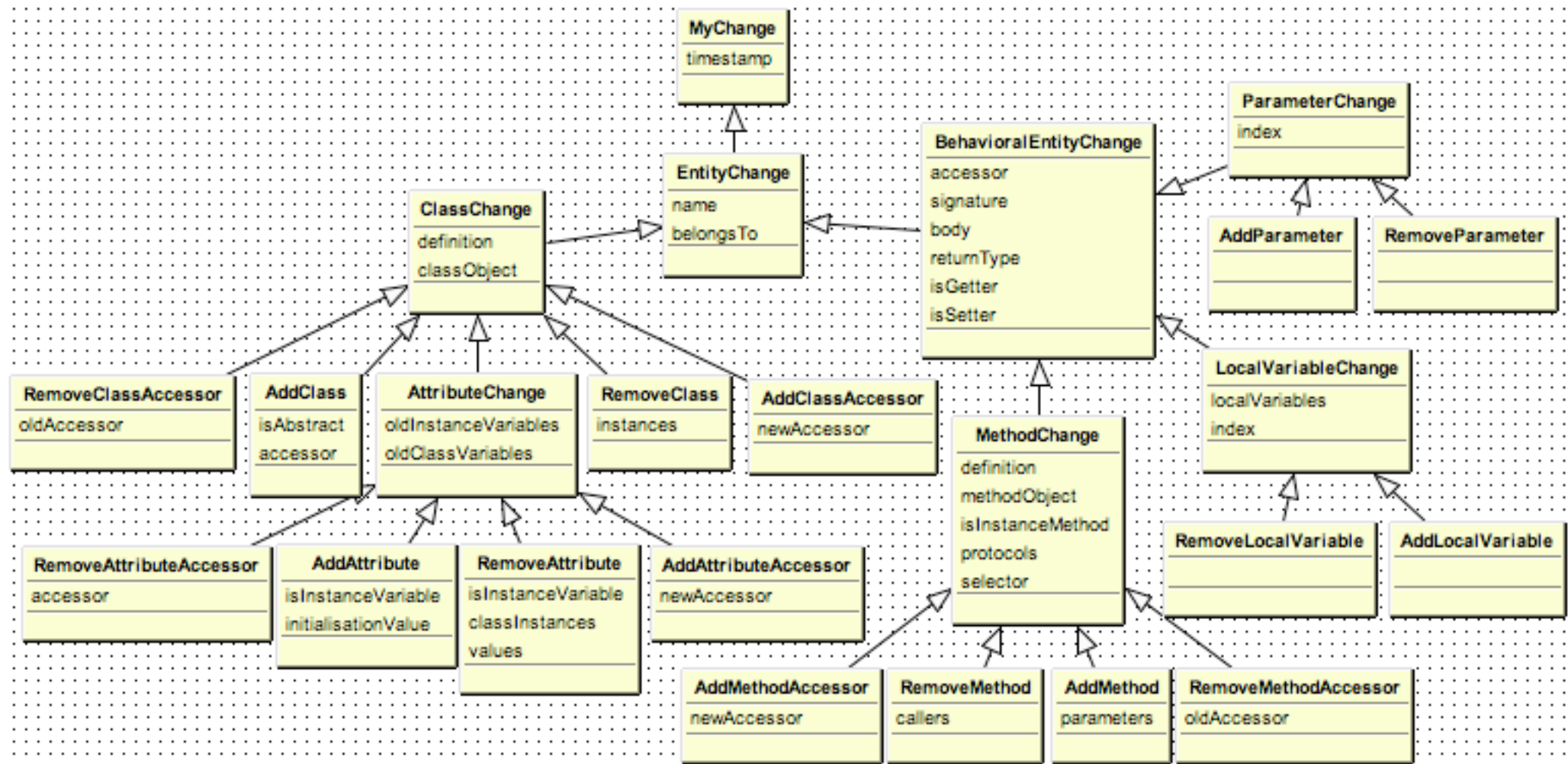
Composable Changes

Dependent Changes

Intensional Changes

Change-oriented IDE

Fine-grained Changes



Model of First-class Changes

Fine-grained Changes

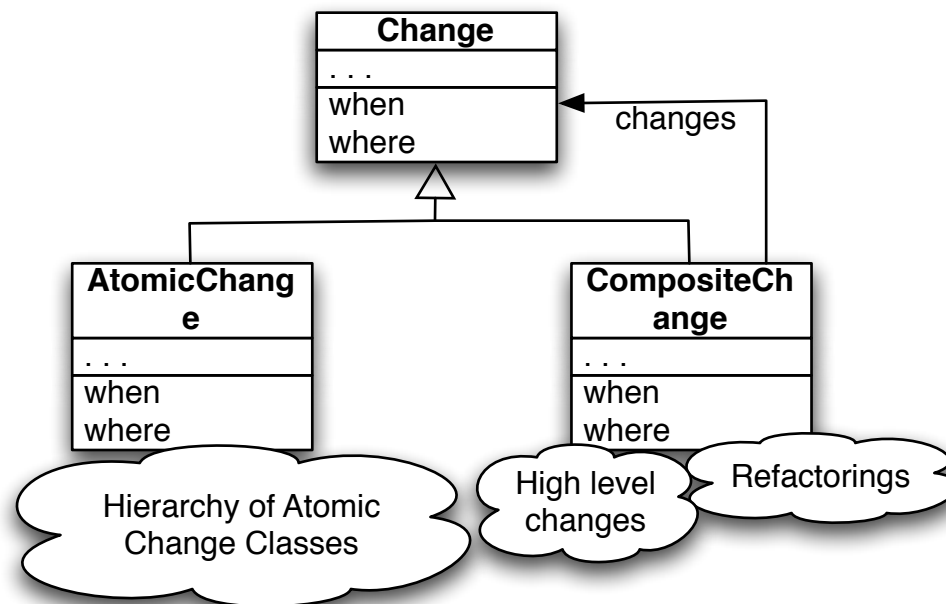
Composable Changes

Dependent Changes

Intensional Changes

Change-oriented IDE

Composable Changes



Model of First-class Changes

Fine-grained Changes

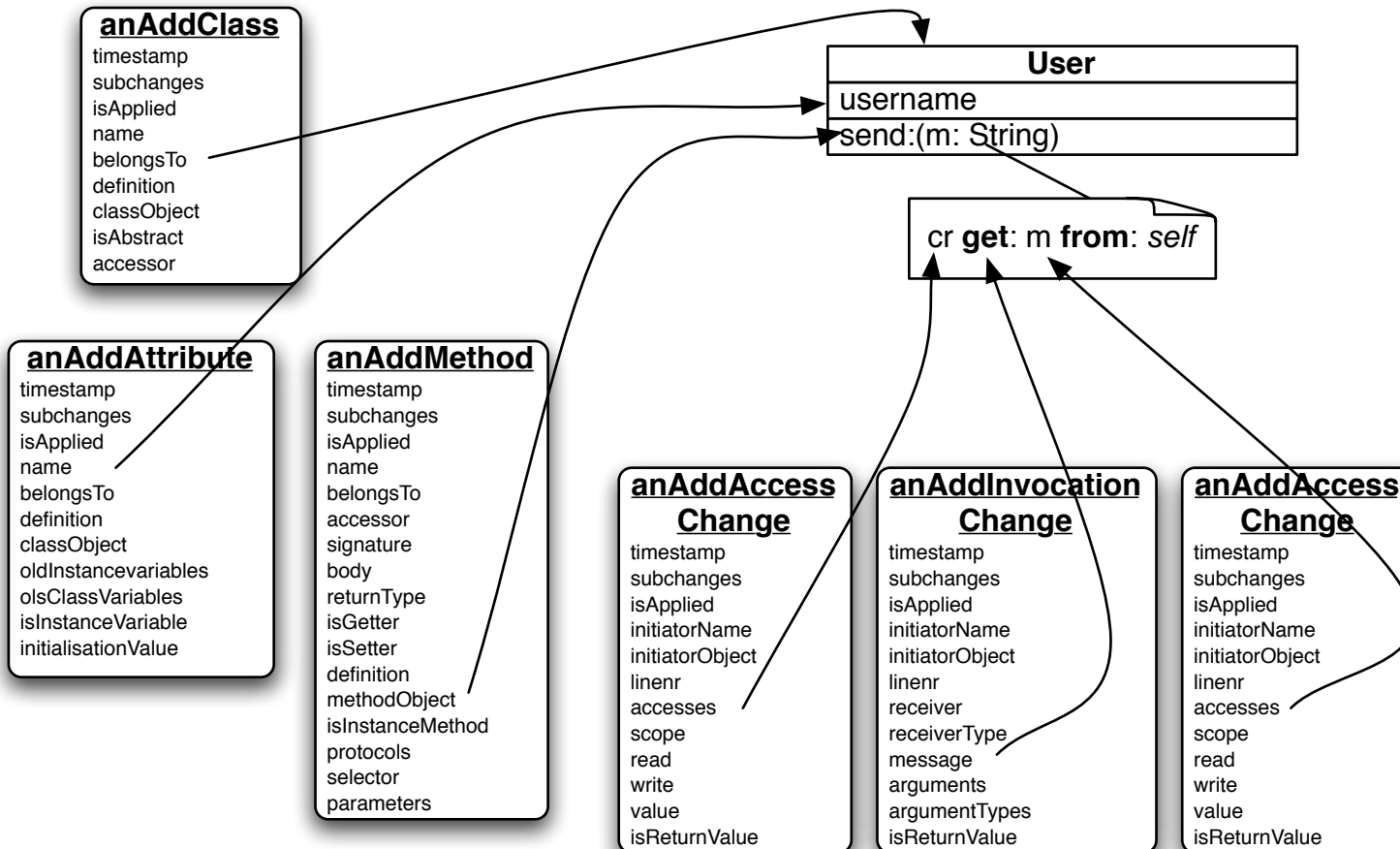
Composable Changes

Dependent Changes

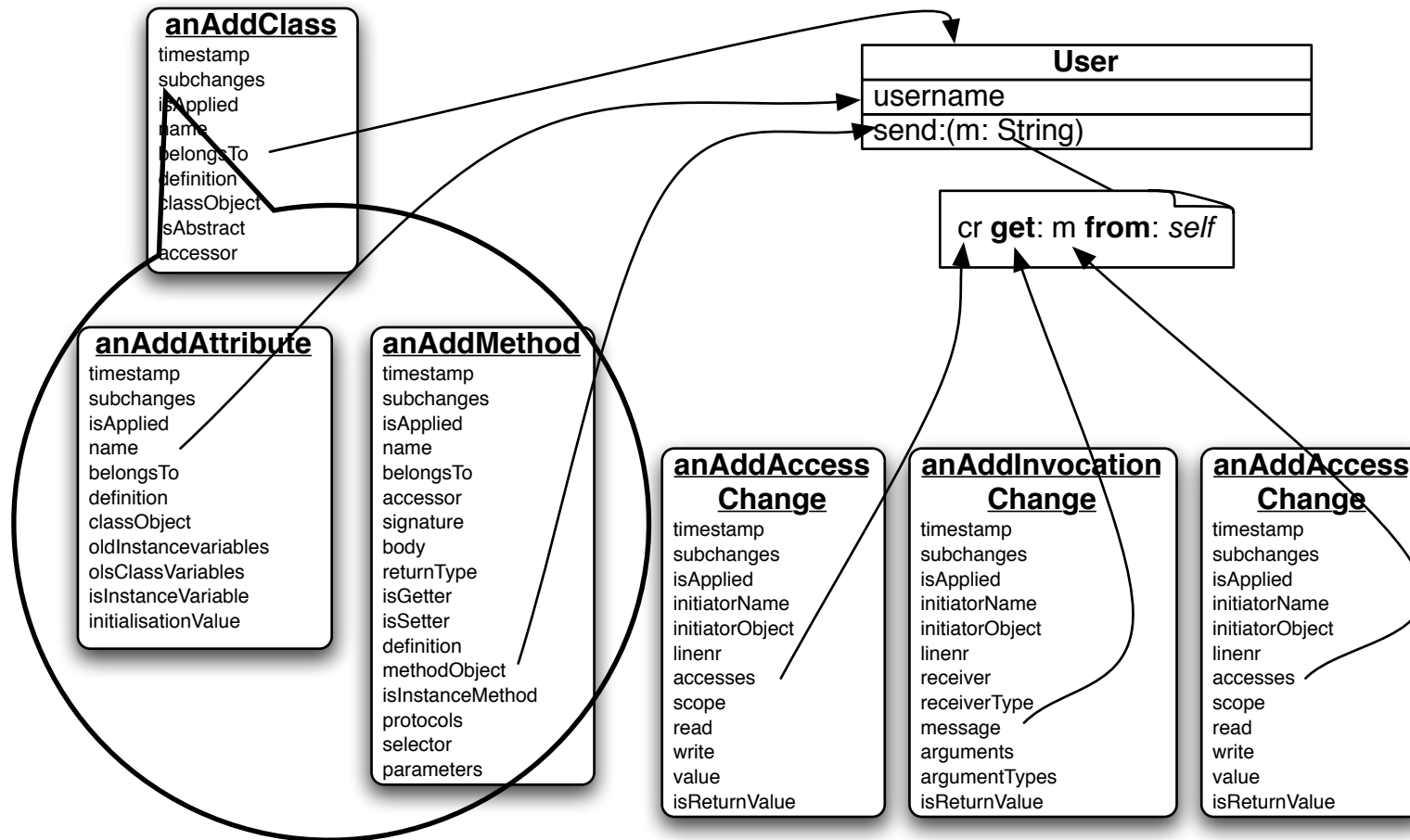
Intensional Changes

Change-oriented IDE

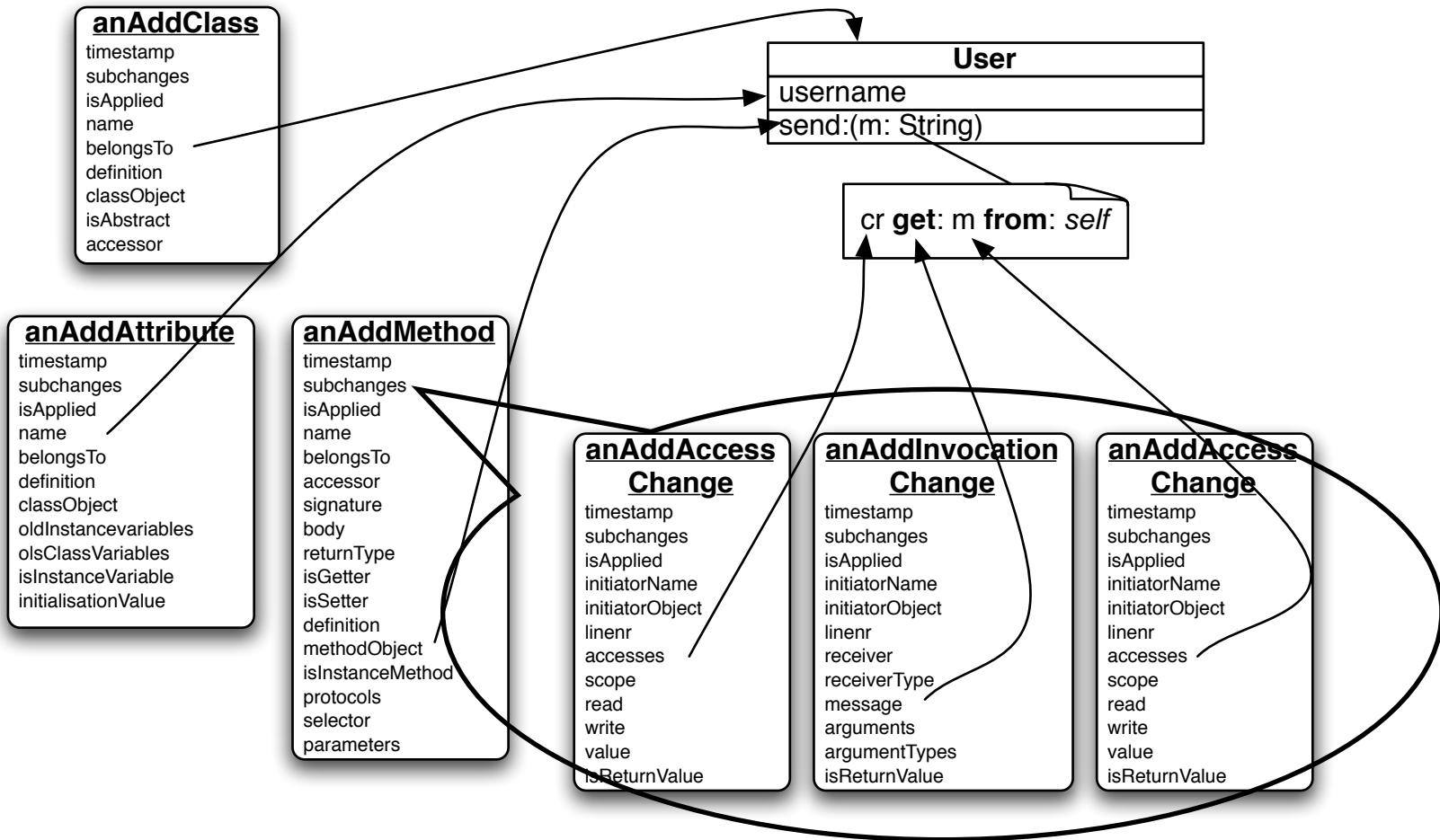
Dependent Changes



Dependent Changes



Dependent Changes



Model of First-class Changes

Fine-grained Changes

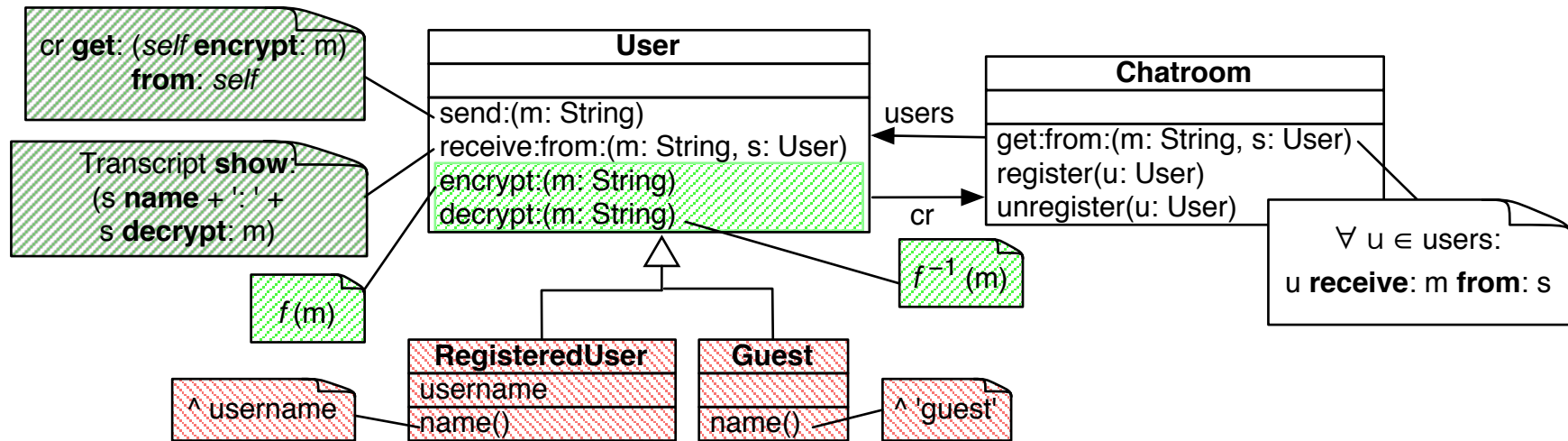
Composable Changes

Dependent Changes

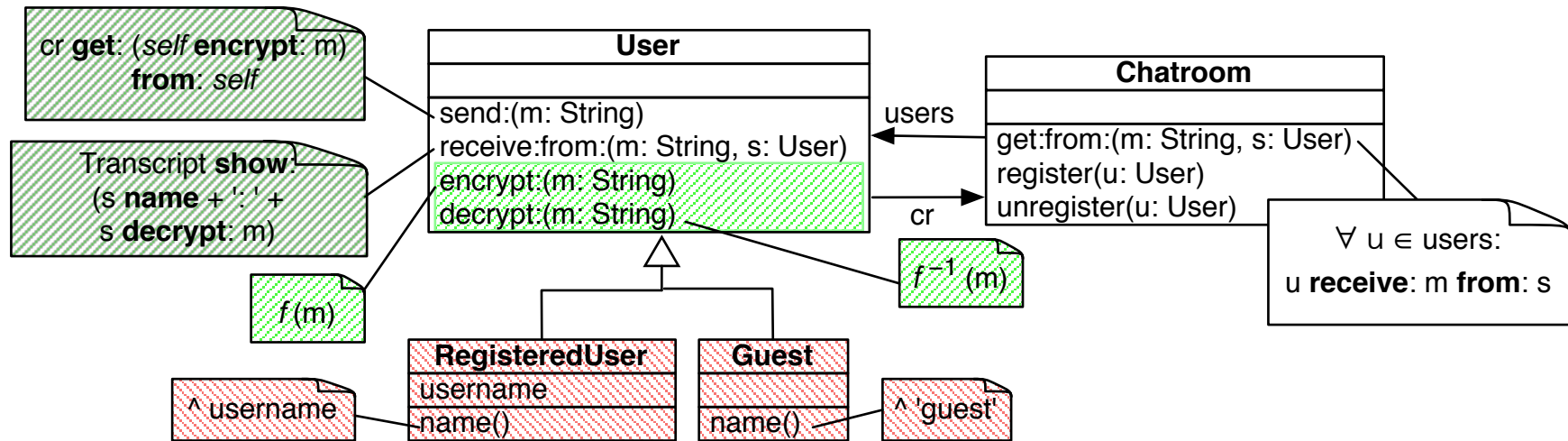
Intensional Changes

Change-oriented IDE

Intensional Changes

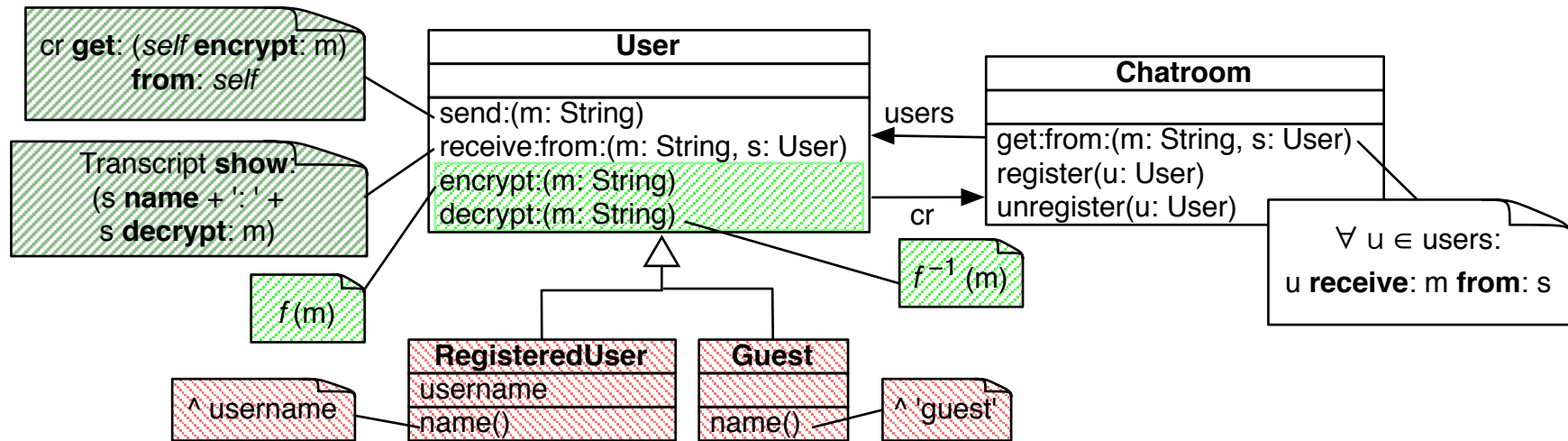


Intensional Changes



- Rename method Refactoring:
 - Change the `name()` method to `username()`

Intensional Changes



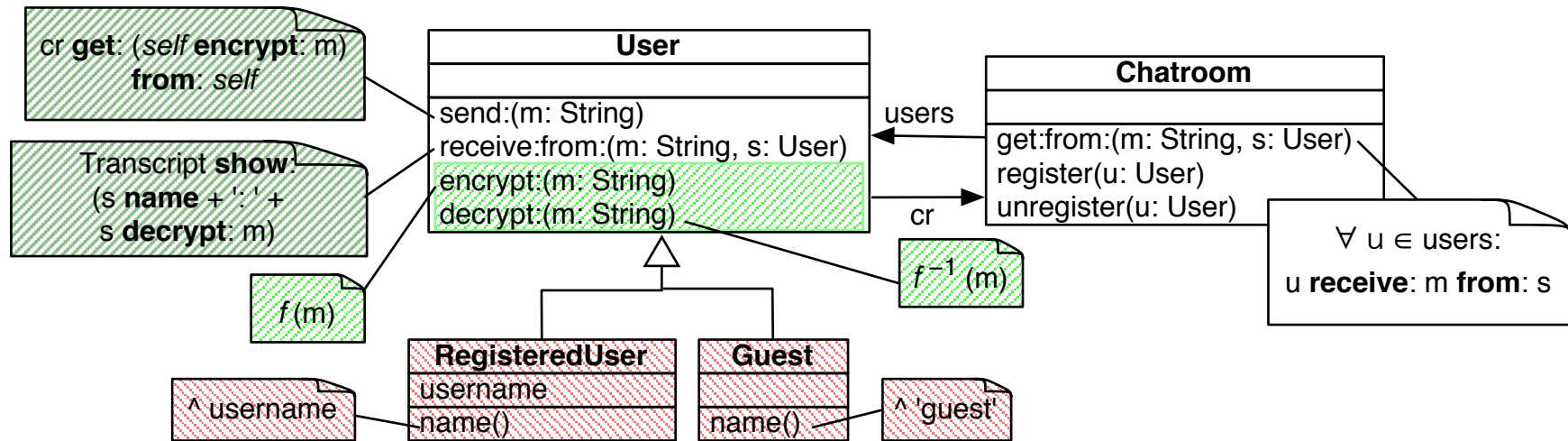
Algorithm 1 Change method body: extension

Change the name of method name in class RegisteredUser to username

Change the name of method name in class Guest to username

Change the invocation s name in method receive:from: from User to s username

Intensional Changes

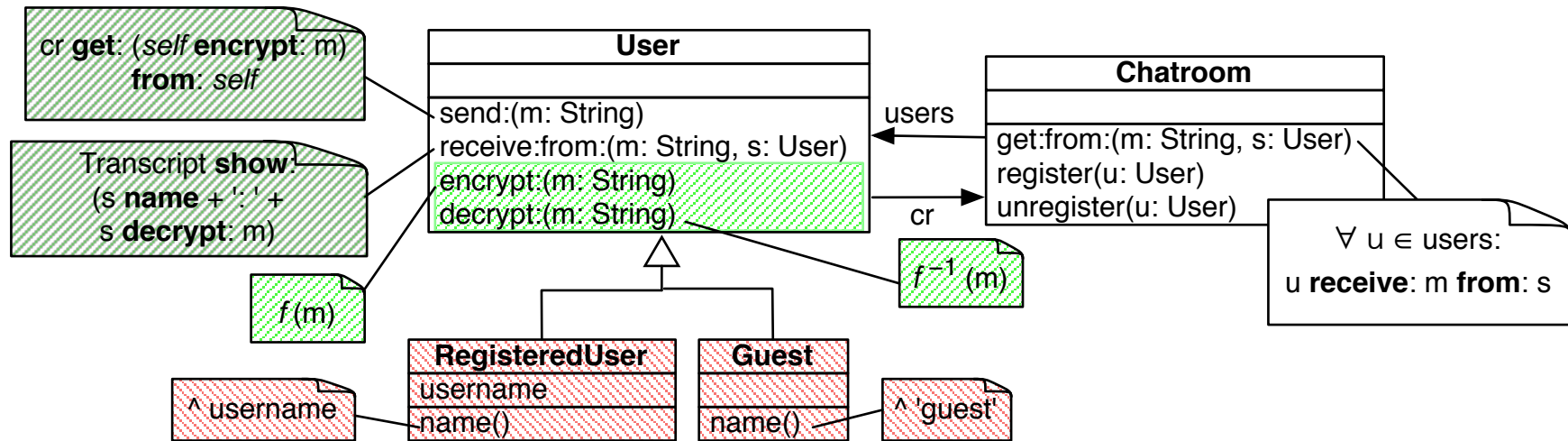


Algorithm 2 Change method body: intension

Change the name of method name in all subclasses of the class User to username

Change every invocation of that method name to an invocation of method username

Intensional Changes



Algorithm 2 Change method body: intension

Change the name of method name in all subclasses of the class User to username

Change every invocation of that method name to an invocation of method username

=> More robust to change

Model of First-class Changes

Fine-grained Changes

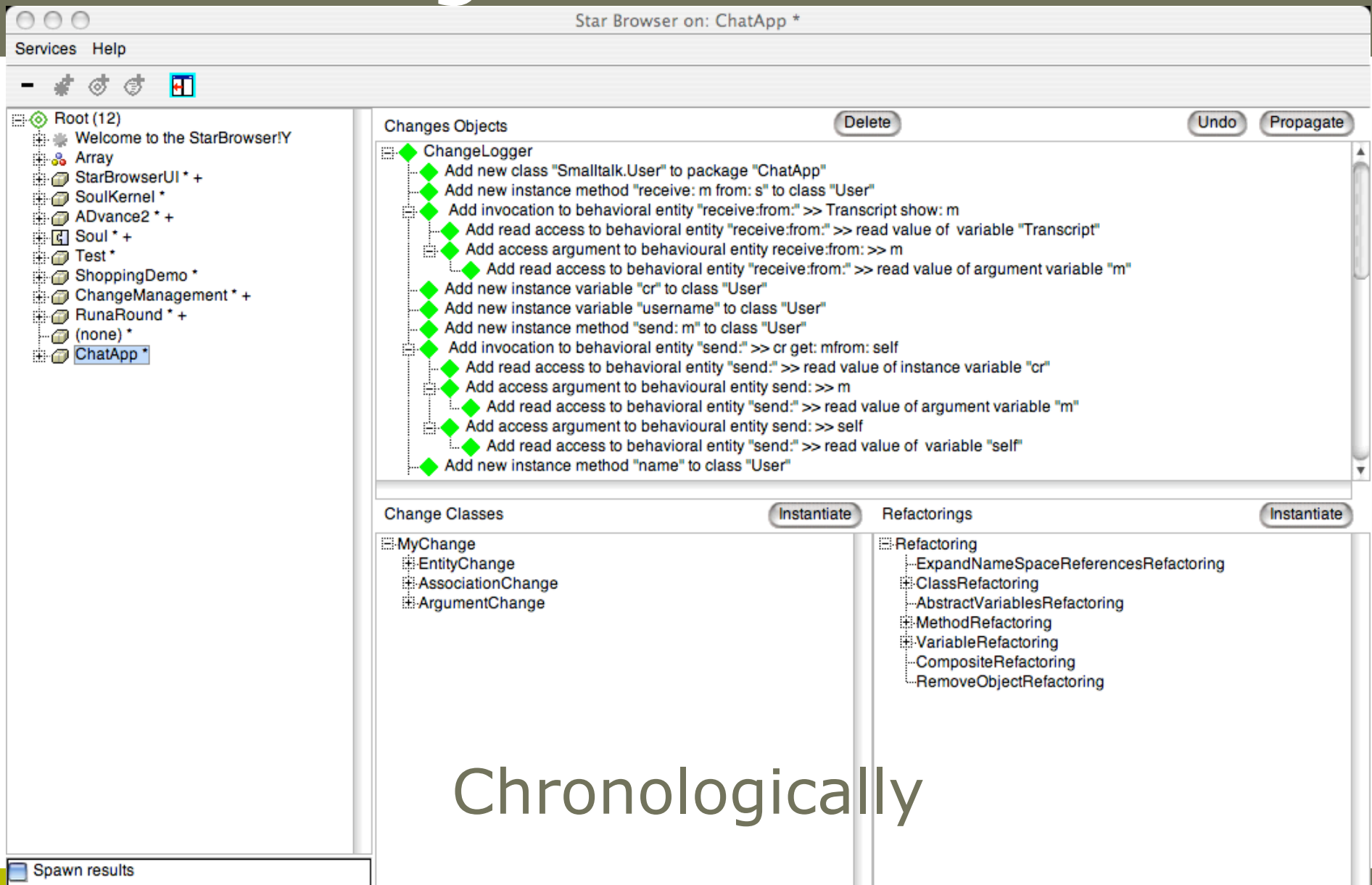
Composable Changes

Dependent Changes

Intensional Changes

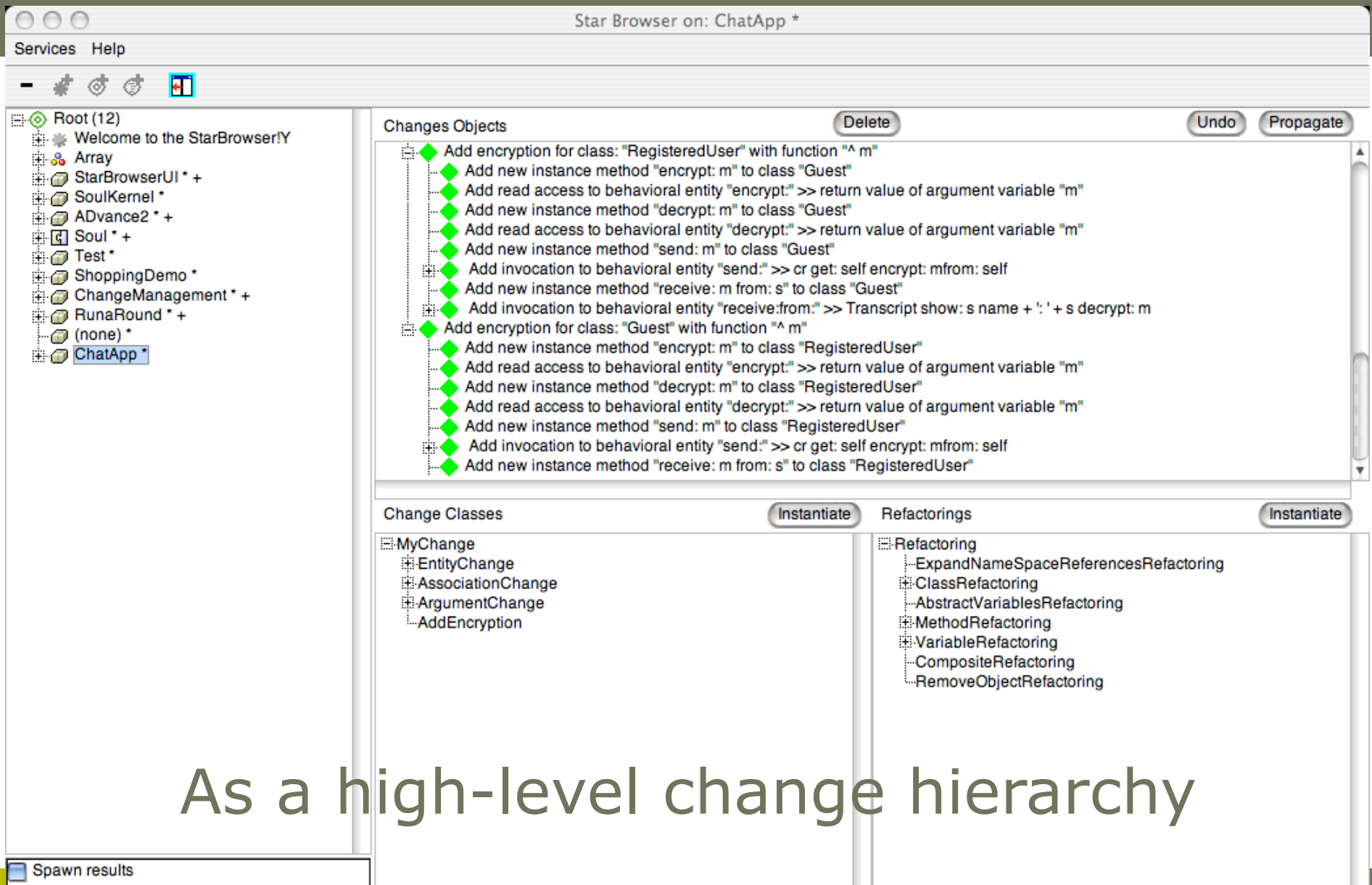
Change-oriented IDE

Change-Oriented IDE



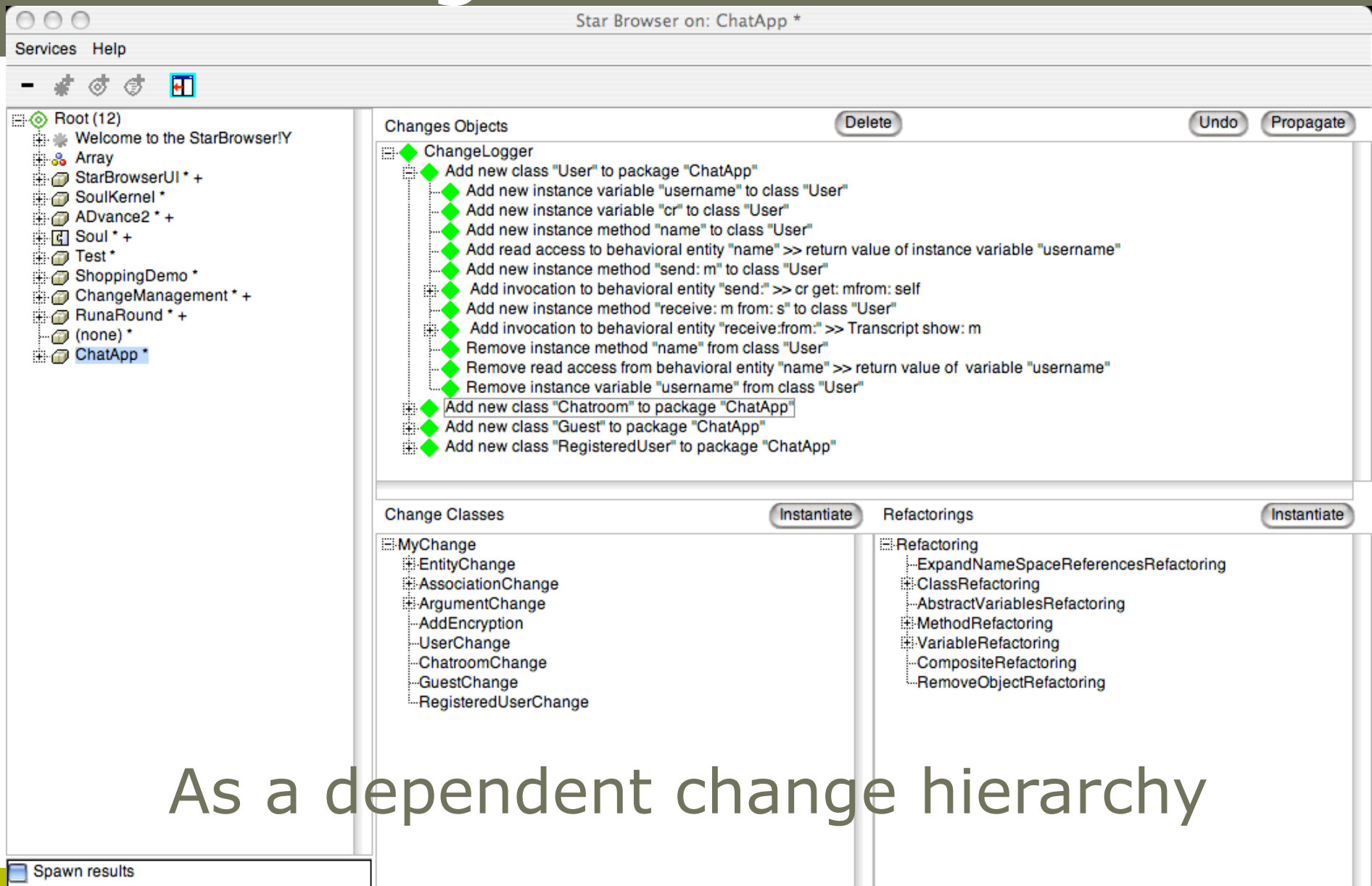
Chronologically

Change-Oriented IDE



As a high-level change hierarchy

Change-Oriented IDE



As a dependent change hierarchy

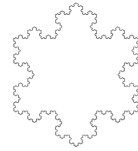
Change List by ChEOPS

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

- 1 Changes on ChatApp package:
- 2 Add new instance method "encrypt: m" to class "User"
- 3 -> Invocation tree Added
- 4 Add new instance method "decrypt: m" to class "User"
- 5 -> Invocation tree Added
- 6 Remove instance method "receive: m from: s" from class "User"
- 7 -> Invocation tree Removed
- 8 Add new instance method "receive: m from: s" to class "User"
- 9 -> Invocation tree Added
- 10 Remove instance method "send: m" from class "User"
- 11 -> Invocation tree Removed
- 12 Add new instance method "send: m" to class "User"
- 13 -> Invocation tree Added

Change List by ChEOPS



Restricted Level Of Granularity

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

- 1 Changes on ChatApp package:
- 2 Add new instance method "encrypt: m" to class "User"
- 3 -> Invocation tree Added
- 4 Add new instance method "decrypt: m" to class "User"
- 5 -> Invocation tree Added
- 6 Remove instance method "receive: m from: s" from class "User"
- 7 -> Invocation tree Removed
- 8 Add new instance method "receive: m from: s" to class "User"
- 9 -> Invocation tree Added
- 10 Remove instance method "send: m" from class "User"
- 11 -> Invocation tree Removed
- 12 Add new instance method "send: m" to class "User"
- 13 -> Invocation tree Added

Change List by ChEOPS



Restricted Level Of Granularity

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

- 1 Changes on ChatApp package:
- 2 Add new instance method "encrypt: m" to class "User"
- 3 -> Invocation tree Added
- 4 Add new instance method "decrypt: m" to class "User"
- 5 -> Invocation tree Added
- 6 Remove instance method "receive: m from: s" from class "User"
- 7 -> Invocation tree Removed
- 8 Add new instance method "receive: m from: s" to class "User"
- 9 -> Invocation tree Added
- 10 Remove instance method "send: m" from class "User"
- 11 -> Invocation tree Removed
- 12 Add new instance method "send: m" to class "User"
- 13 -> Invocation tree Added

Change List by ChEOPS

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

Restricted Level Of Granularity

Term Overloading



- 1 Changes on ChatApp package:
- 2 Add new instance method "encrypt: m" to class "User"
- 3 -> Invocation tree Added
- 4 Add new instance method "decrypt: m" to class "User"
- 5 -> Invocation tree Added
- 6 Remove instance method "receive: m from: s" from class "User"
- 7 -> Invocation tree Removed
- 8 Add new instance method "receive: m from: s" to class "User"
- 9 -> Invocation tree Added
- 10 Remove instance method "send: m" from class "User"
- 11 -> Invocation tree Removed
- 12 Add new instance method "send: m" to class "User"
- 13 -> Invocation tree Added

Change List by ChEOPS

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

Restricted Level Of Granularity

Term Overloading



- 1 Changes on ChatApp package:
- 2 Add new instance method "encrypt: m" to class "User"
- 3 -> Invocation tree Added
- 4 Add new instance method "decrypt: m" to class "User"
- 5 -> Invocation tree Added
- 6 Remove instance method "receive: m from: s" from class "User"
- 7 -> Invocation tree Removed
- 8 Add new instance method "receive: m from: s" to class "User"
- 9 -> Invocation tree Added
- 10 Remove instance method "send: m" from class "User"
- 11 -> Invocation tree Removed
- 12 Add new instance method "send: m" to class "User"
- 13 -> Invocation tree Added

Change List by ChEOPS

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

Restricted Level Of Granularity

Term Overloading

Lack Of High-level Changes



- 1 Changes on ChatApp package:
- 2 Add new instance method "encrypt: m" to class "User"
- 3 -> Invocation tree Added
- 4 Add new instance method "decrypt: m" to class "User"
- 5 -> Invocation tree Added
- 6 Remove instance method "receive: m from: s" from class "User"
- 7 -> Invocation tree Removed
- 8 Add new instance method "receive: m from: s" to class "User"
- 9 -> Invocation tree Added
- 10 Remove instance method "send: m" from class "User"
- 11 -> Invocation tree Removed
- 12 Add new instance method "send: m" to class "User"
- 13 -> Invocation tree Added

Change List by ChEOPS

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

Restricted Level Of Granularity

Term Overloading

Lack Of High-level Changes



- 1 Changes on ChatApp package:
- 2 Add encryption for class "RegisteredUser" with function "f(x)= encrypt(x)" and "f(x)= decrypt(x)"
- 3 -> composite changes
- 4 Add encryption for class "Guest" with function "f(x)=x" and "f(x)=x"
- 5 -> composite changes

Change List by ChEOPS

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

Restricted Level Of Granularity

Term Overloading

Lack Of High-level Changes



- 1 Changes on ChatApp package:
- 2 Add encryption for class "RegisteredUser" with function "f(x)= encrypt(x)" and "f(x)= decrypt(x)"
- 3 -> composite changes
- 4 Add encryption for class "Guest" with function "f(x)=x" and "f(x)=x"
- 5 -> composite changes

Change List by ChEOPS

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

Restricted Level Of Granularity

Term Overloading

Lack Of High-level Changes

No exploration facilities



- 1 Changes on ChatApp package:
- 2 Add encryption for class "RegisteredUser" with function "f(x)= encrypt(x)" and "f(x)= decrypt(x)"
- 3 -> composite changes
- 4 Add encryption for class "Guest" with function "f(x)=x" and "f(x)=x"
- 5 -> composite changes

Change List by ChEOPS

- 1 Changes on ChatApp package: [vegsoc.gif](#)
- 2 Add new class "User"
- 3 Add new instance method "receive: m from: s" to class "User"
- 4 -> Invocation tree
- 5 Add new instance variable "cr" to class "User"
- 6 Add new instance variable "username" to class "User"
- 7 Add new instance method "send: m" to class "User"
- 8 -> Invocation tree
- 9 Add new instance method "name" to class "User"
- 10 -> Add Read Access
- 11 Add new class "ChatRoom"
- 12 Add new instance variable "users" to class "ChatRoom"
- 13 Add new instance method "get: m from: s" to class "ChatRoom"
- 14 -> Invocation tree
- 15 Add new instance method "register: u" to class "ChatRoom"
- 16 -> Invocation tree
- 17 Add new instance method "unregister: u" to class "ChatRoom"
- 18 Add invocation "users remove: u"

- 1 Changes on ChatApp package:
- 2 Add new class "Guest"
- 3 Add new instance method "name" to class "Guest"
- 4 -> Invocation tree Added
- 5 Add new class "RegisteredUser"
- 6 Add new empty instance method "name" to class "RegisteredUser"
- 7 Add read access to behavioral entity "name" >> return value of variable "username"
- 8 Remove instance method "name" from class "User"
- 9 -> Invocation tree Removed

Restricted Level Of Granularity

Term Overloading

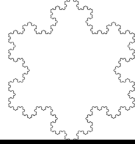



Lack Of High-level Changes

No exploration facilities



- 1 Changes on ChatApp package:
- 2 Add encryption for class "RegisteredUser" with function "f(x)= encrypt(x)" and "f(x)= decrypt(x)"
- 3 -> composite changes
- 4 Add encryption for class "Guest" with function "f(x)=x" and "f(x)=x"
- 5 -> composite changes

Discussion

Extension				
Fine-grained changes	v	v		
Composable changes	v		v	v
Dependent changes				v
Intensional changes			v	v
IDE support			v	v

Conclusion

- Support for Change-Oriented Software Engineering
- Current models of first-class changes are insufficient:
 - The **restricted level of granularity** in the different types of changes
 - The **overloading** of change types
 - The **lack of high-level** changes
 - The **lack of program exploration** facilities
- Extensions to the model overcome those issues
- Change- and Evolution-Oriented Programming Support

Future Work

- COSE for component-oriented programming
- Provide other semantics to the changes
 - Platform-specific code maintenance based on the changes
- Maintain the “raison d’être” of program entities
- Improved IDE support for COSE