

Generative Programming from a Post Object-Oriented Programming Viewpoint

Shigeru Chiba

Tokyo Institute of Technology

[Craftsmen]

- Men with great skill crafts great things.



Bamboo fishing rod



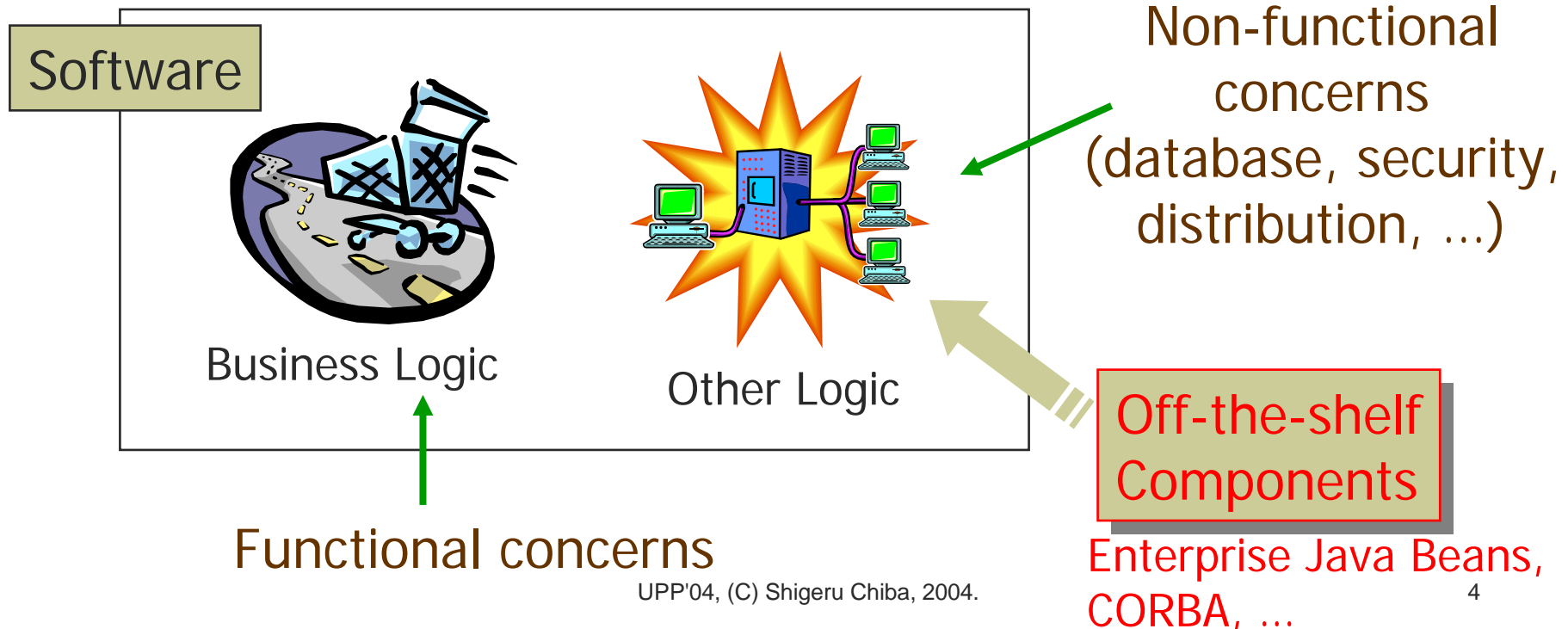
[Modern Industry]

- Quality might be so-so but...

Photo
(Industrial robots that are constructing an automobile)

Off-the-shelf components

- Reduces software development costs



Hidden costs of using components

- Complicated programming frameworks need craftsman.
 - Developers must read a thick manual book.

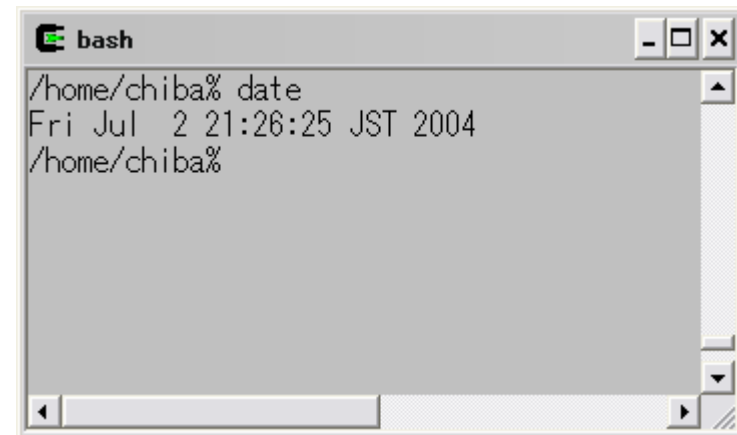


Skilled engineers can easily get a job.

[e.g. Developing a clock]

- Simple program

```
void main() {  
    while (true) {  
        Draw the time;  
        wait for a minute;  
    }  
}
```

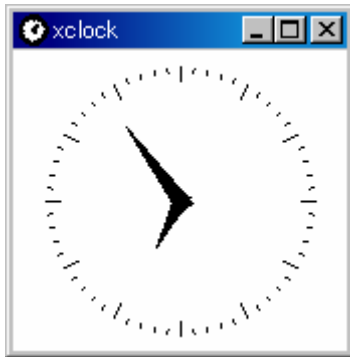


A screenshot of a terminal window titled 'bash'. The prompt is '/home/chiba%'. The user has entered 'date', and the output is 'Fri Jul 2 21:26:25 JST 2004'. The prompt is now '/home/chiba%'. The terminal window has standard window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

- Use components for better look!

[Use components]

- Must rewrite a program to fit the framework
 - The program must be a subclass of Applet.



```
class Clock extends Applet {  
    void main() {  
        while (true) {  
            Draw the time;  
            wait for a minute;  
        }  
    }  
}
```

[Use components]

- Must rewrite a program to fit the framework
 - The program must extend Applet
 - paint() must draw the clock



```
class Clock extends Applet {  
    void paint() {  
        Draw the time;  
    }  
    void main() {  
        while (true) {  
            paint();  
            wait for a minute;  
        }  
    }  
}
```


Use components

- Must rewrite a p
 - The program m
 - paint() must dra
 - Implement the F



```
class Clock extends Applet
    implements Runnable {
    void paint() {
        Draw the time; }
    void run() {
        while (true) {
            paint();
            wait for a minute;
        }
    }
    void main() {
        new Thread(...).start();
    }
}
```

[A new component framework]

- Learning it is like learning a new language.
- Our EJB 2.0 developers need to have expensive training for EJB 3.0.

- An IT Manager

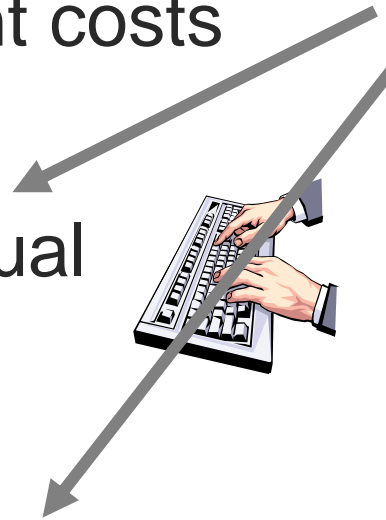


People from training business

Automate this rewriting process

- Easy to use components
 - Reduce development costs
- Now,
 - English/French manual
 - Rewrite by human
- Research goal,
 - Machine-readable description
 - Rewrite by computer

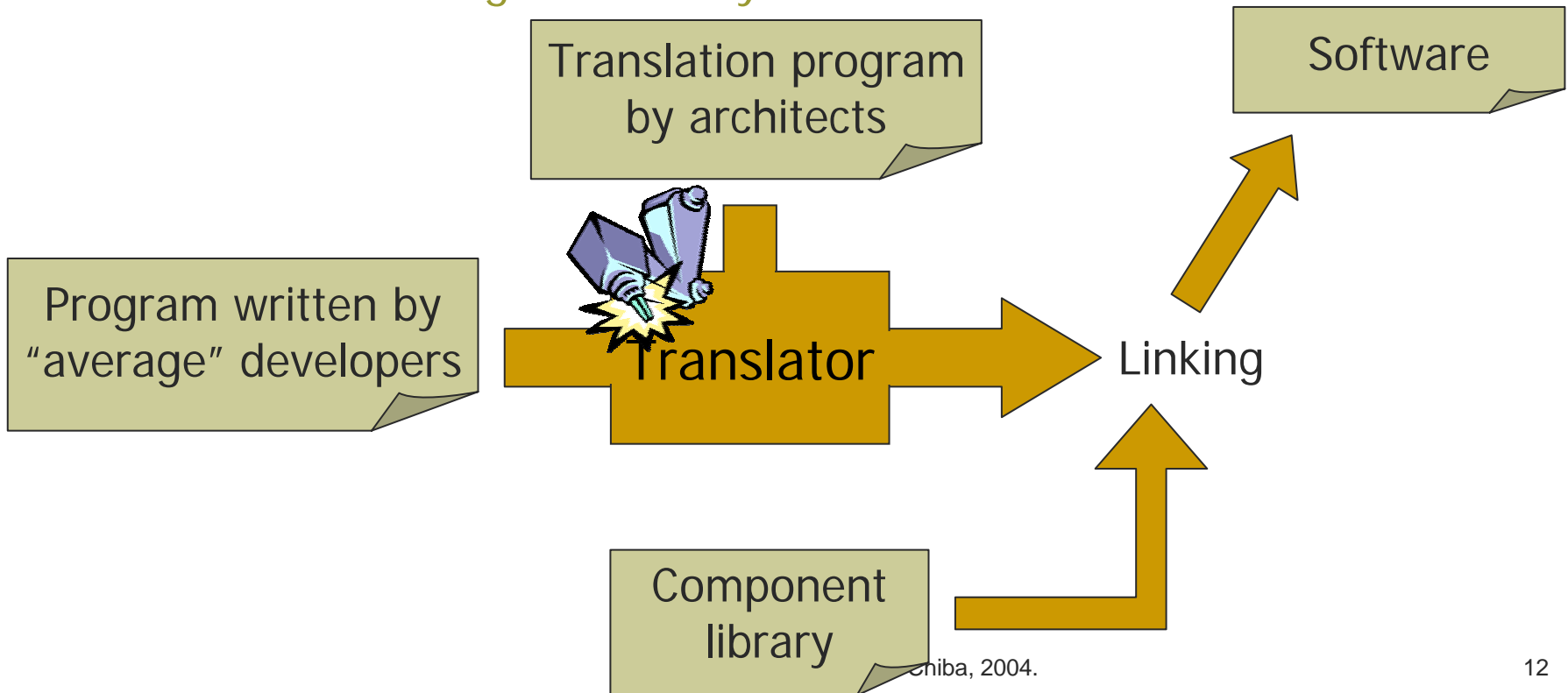
Provided by
component
developers



Generative Programming

■ Programmable Program Translator

High reusability



[Key Technologies]

- How do you describe translation?
 - Meta programming (Reflection)
 - Model-Driven Architecture
 - Aspect-oriented Programming

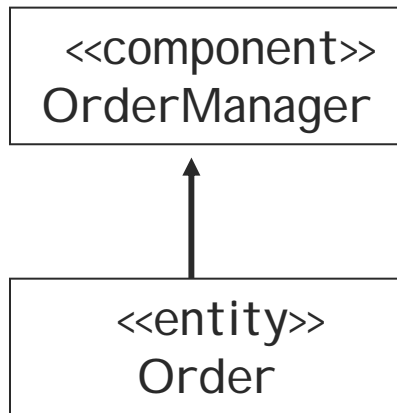
[Meta Programming]

- OpenC++ and Javassist [Chiba95,00]
 - (fairly) production-quality (> K-downloads)
 - Show the structural view of a program.
 - Class, Method, Field, ...
 - A translation program edits those entities.
 - E.g.
 - Change an interface if a class is
 - Appends a method if ...
 - Define a proxy class if ...

Model Driven Architecture (MDA)

■ PIM (Platform Independent Model)

(Warehouse management system)

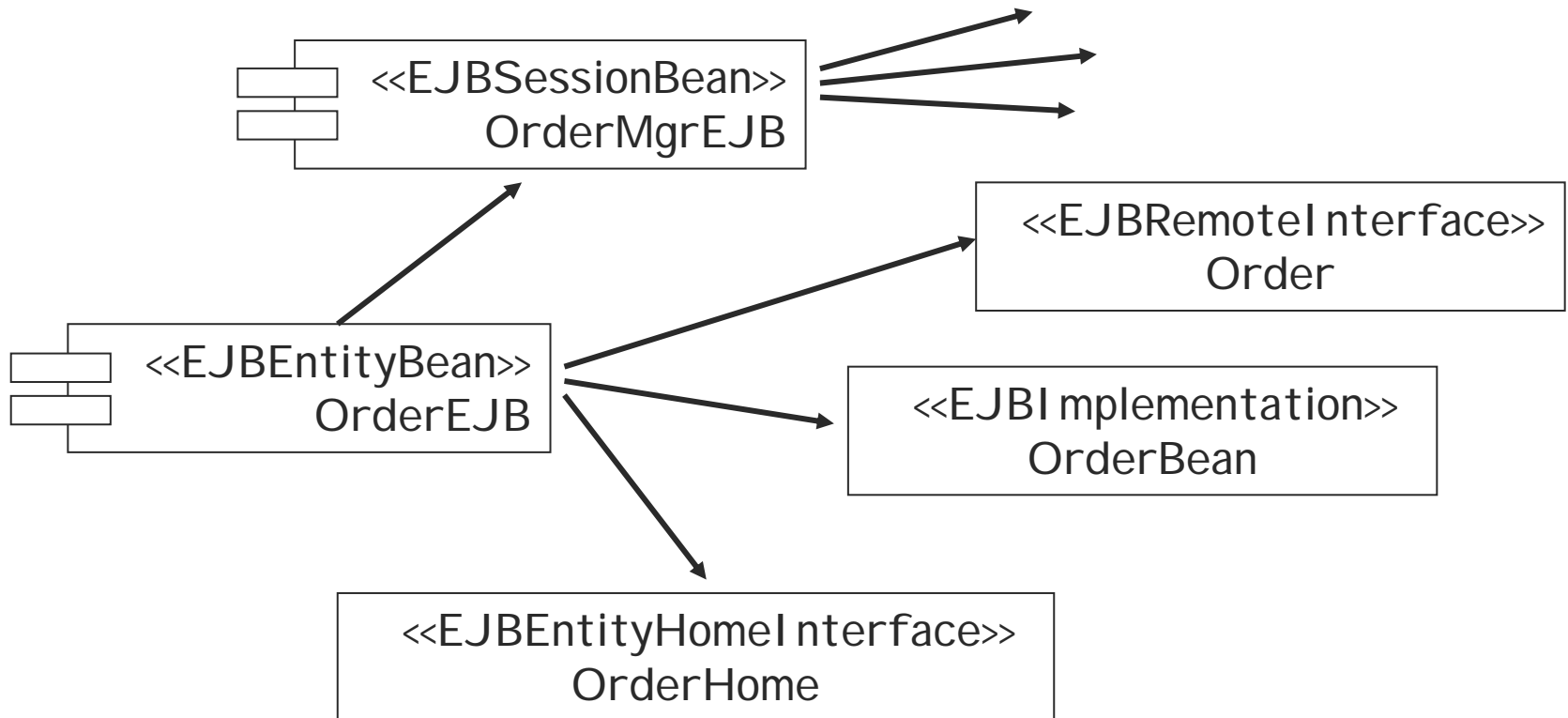


A MDA tool transforms

- Marked entities, or
- Entities matching **some meta model**

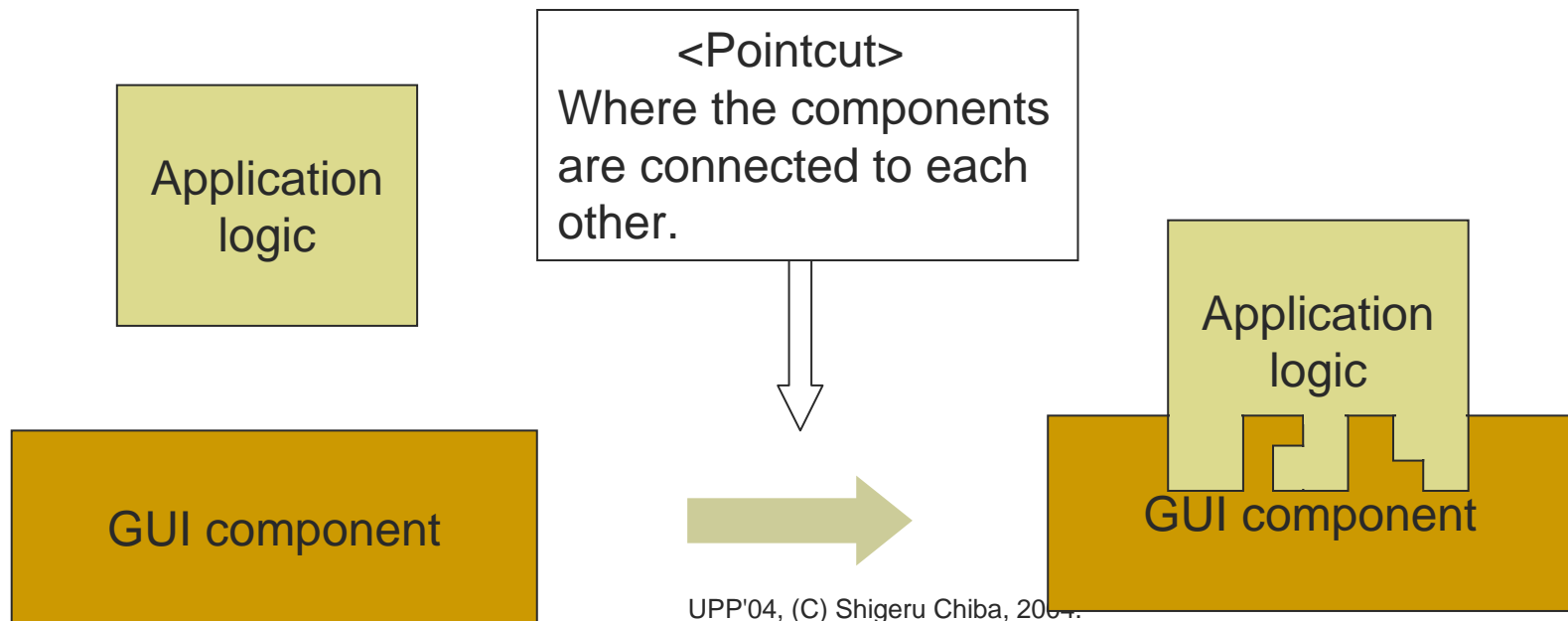
Model Driven Architecture (MDA)

- PIM to PSM (Platform Specific Model)



Aspect-Oriented Programming

- Allows automatically combining programs written on different aspects.



[Summary]

- Automate rewriting code for using to an object-oriented component framework.
- An interesting application of Generative programming