# Computer Architecture Group

**Professors:**
- Antônio Augusto Fröhlich
- Rafael Luiz Cancian

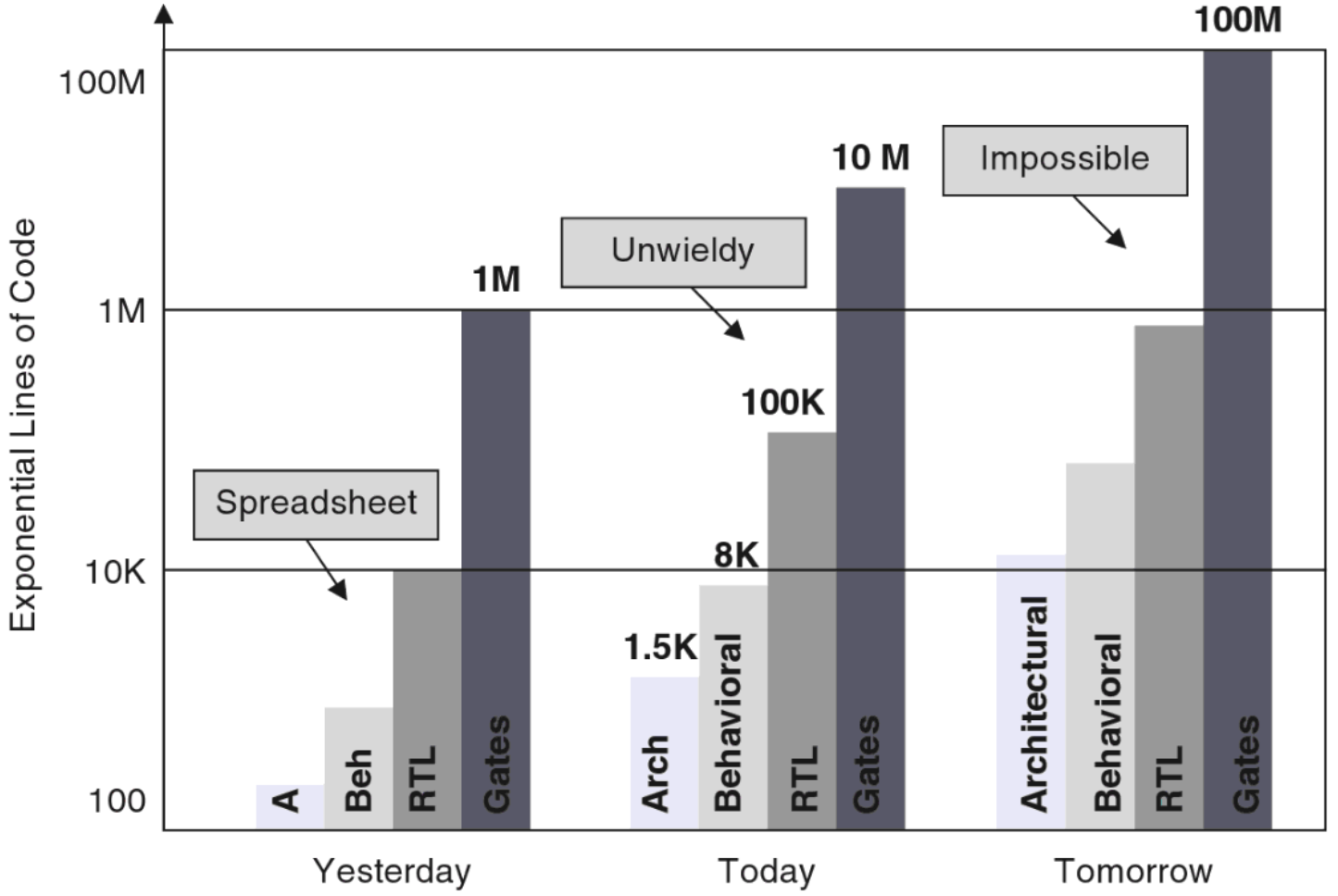
**PhD students:**
- Marcelo Berejuck
- Mateus Krepsky Ludwich

**Msc students:**
- Tiago Rogério Mück
- Wesley Gonçalves

# Introduction

- The complexity systems is always increasing

Code complexity for SoCs designs in various level of abstraction

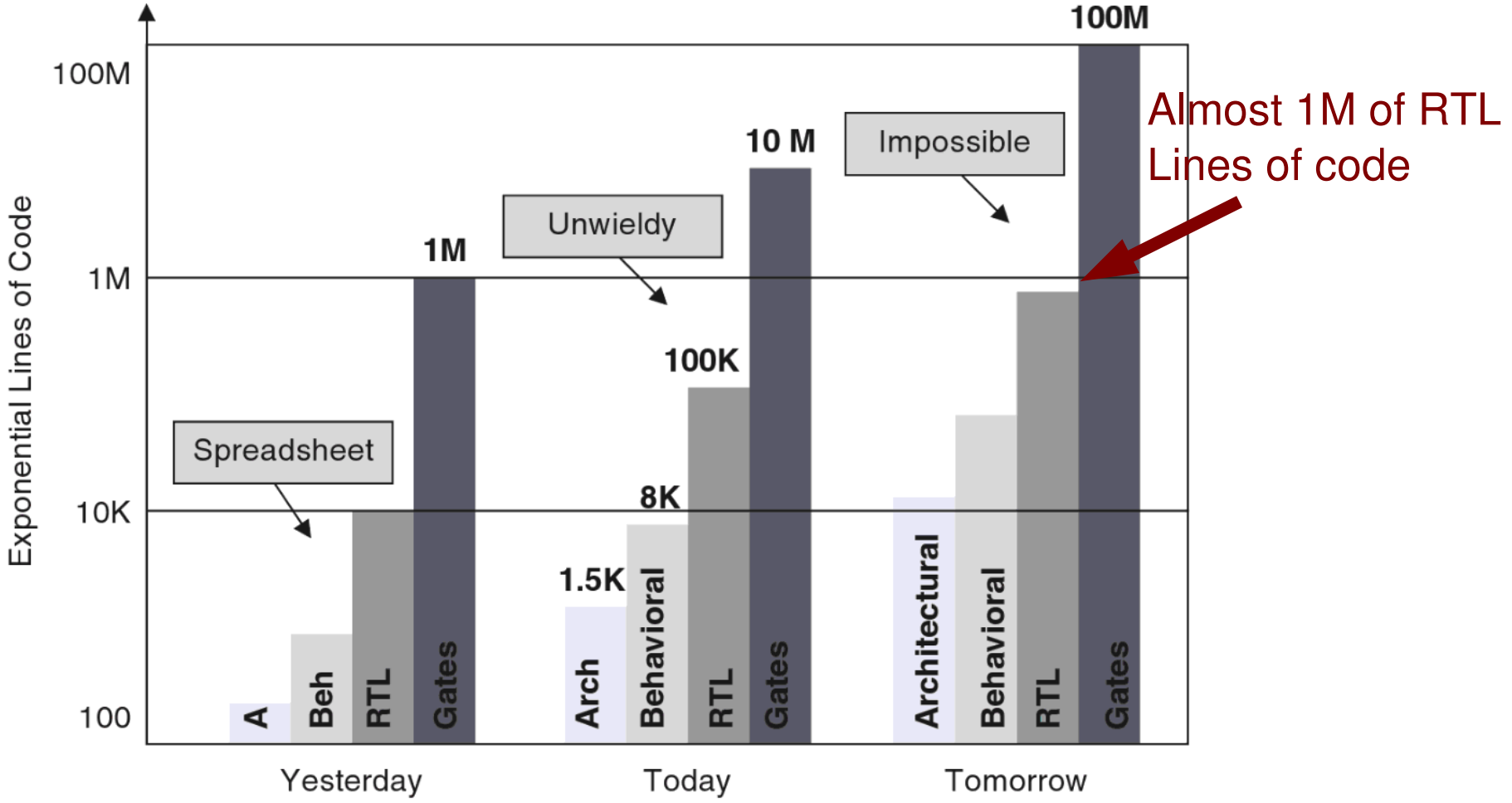



(Black & Donovan, 2009)

# Introduction

- The complexity systems is always increasing

Code complexity for SoCs designs in various level of abstraction

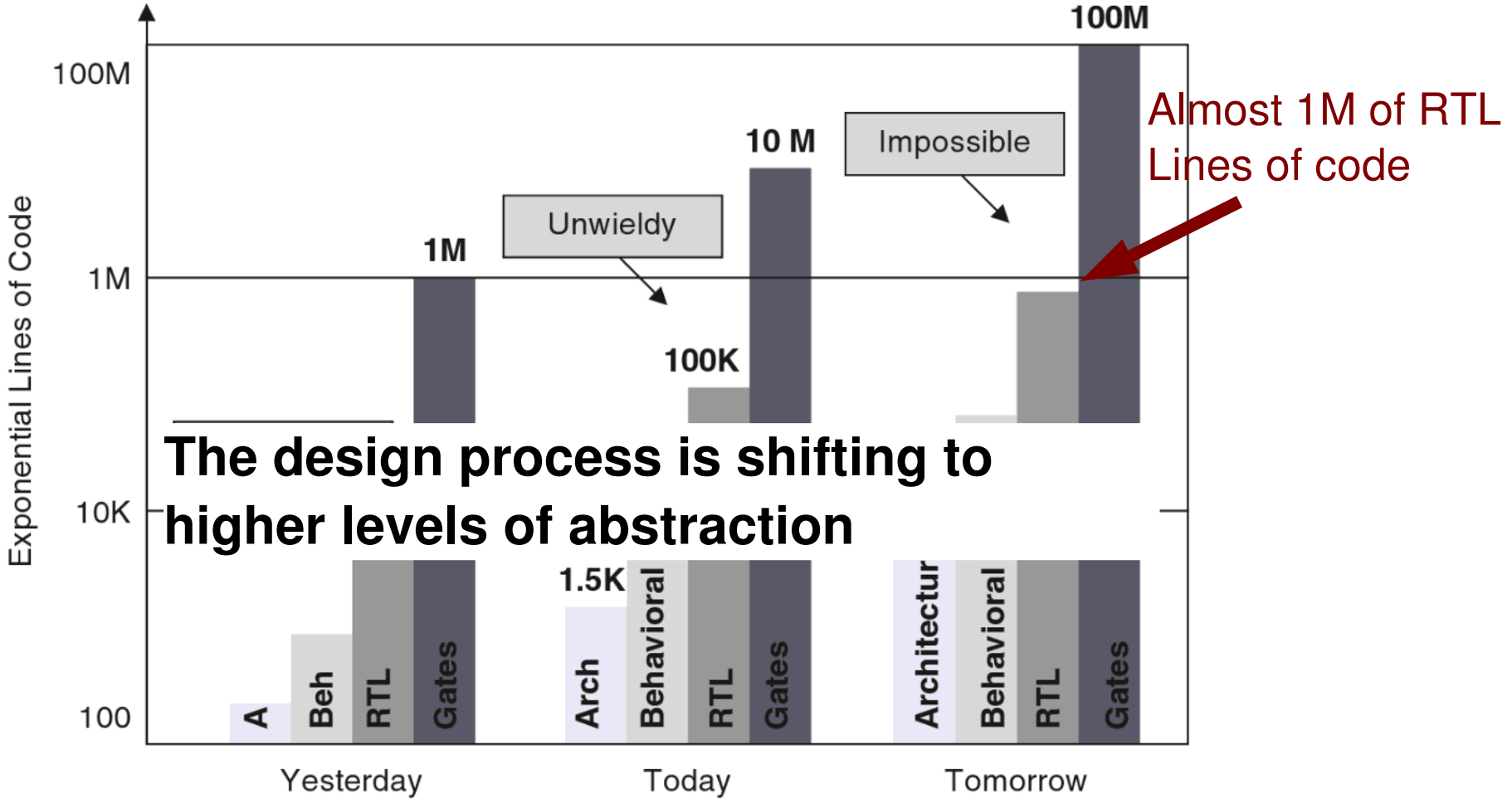



(Black & Donovan, 2009)

# Introduction

- The complexity systems is always increasing

Code complexity for SoCs designs in various level of abstraction




(Black & Donovan, 2009)

# Evolution of abstraction levels

# Evolution of abstraction levels


- For software:

```
ADDI $SP, $SP, -8
SW   $A0, 4($SP)
SW   $RA, 0($SP)
```

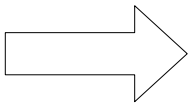
Machine language

# Evolution of abstraction levels

- For software:

```
ADDI $SP, $SP, -8
SW   $A0, 4($SP)
SW   $RA, 0($SP)
```

Machine language

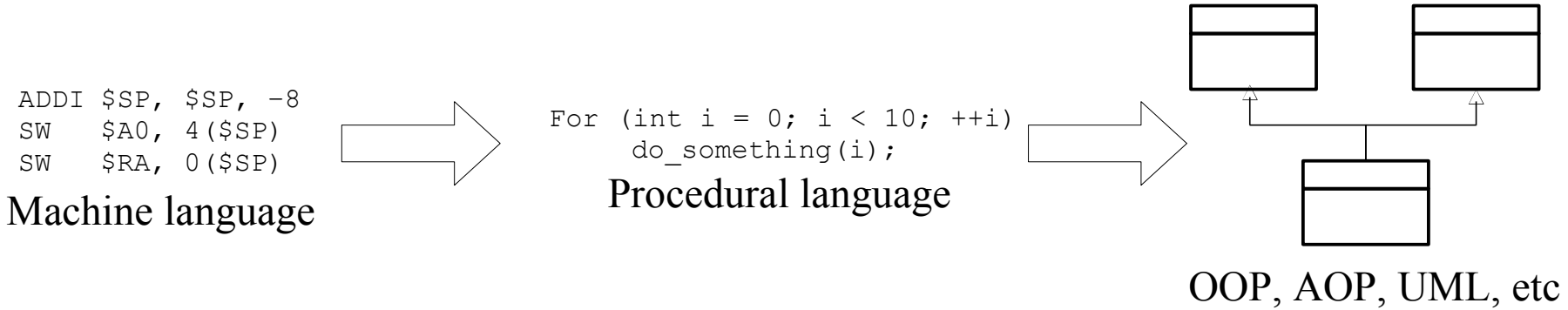


```
For (int i = 0; i < 10; ++i)
     do_something(i);
```

Procedural language

# Evolution of abstraction levels

- For software:



# Evolution of abstraction levels

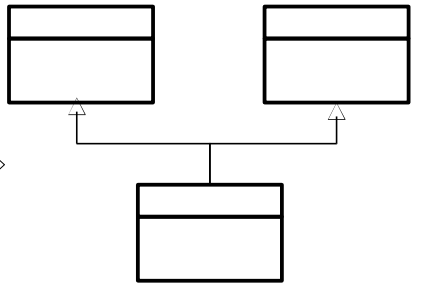
- For software:

```
ADDI $SP, $SP, -8
SW   $A0, 4($SP)
SW   $RA, 0($SP)
```

Machine language

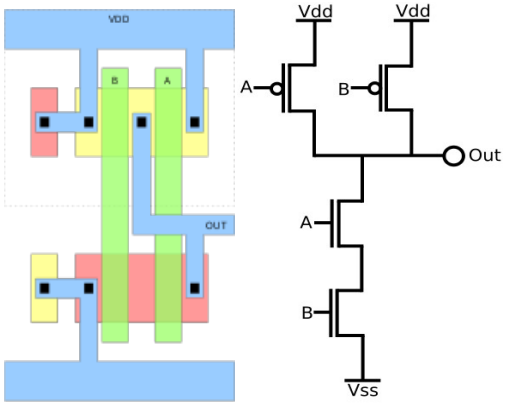
```
For (int i = 0; i < 10; ++i)
    do_something(i);
```

Procedural language



OOP, AOP, UML, etc

- For hardware:




Electric level

# Evolution of abstraction levels

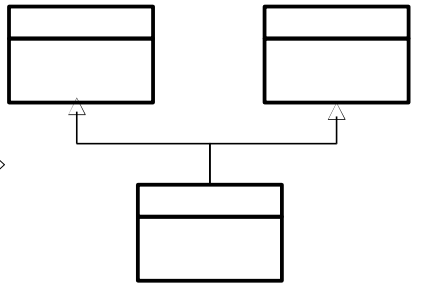
- For software:

```
ADDI $SP, $SP, -8
SW   $A0, 4($SP)
SW   $RA, 0($SP)
```

Machine language

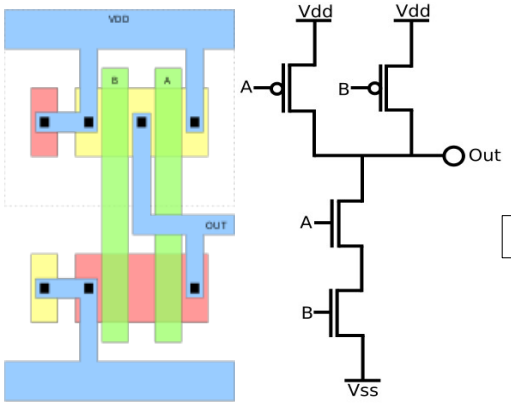
```
For (int i = 0; i < 10; ++i)
    do_something(i);
```

Procedural language



OOP, AOP, UML, etc

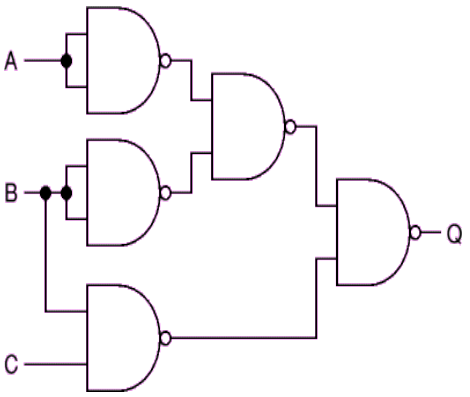
- For hardware:







Electric level




Gate level

# Evolution of abstraction levels

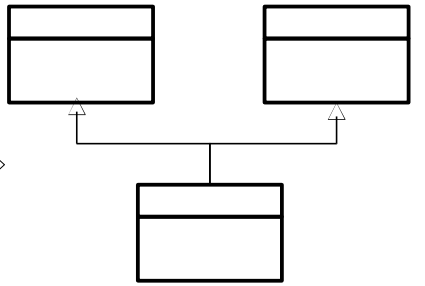
- For software:

```
ADDI $SP, $SP, -8
SW   $A0, 4($SP)
SW   $RA, 0($SP)
```

Machine language

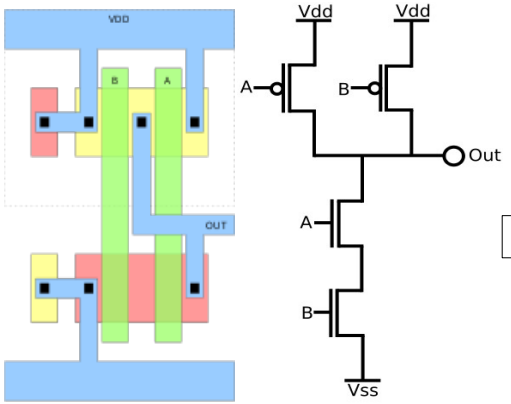
```
For (int i = 0; i < 10; ++i)
    do_something(i);
```

Procedural language



OOP, AOP, UML, etc

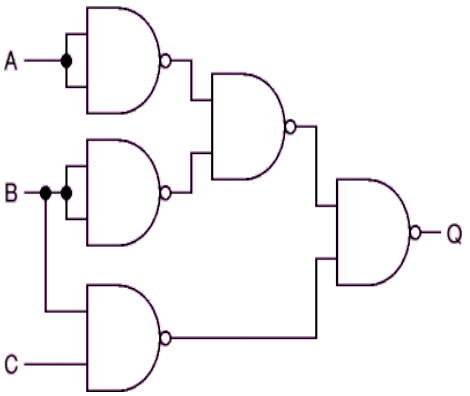
- For hardware:







Electric level




Gate level

```
always @(sel or a or b)
begin
  if (sel == 1)
    f = a;
  else
    f = b;
end
```

Register transfer level

# Evolution of abstraction levels

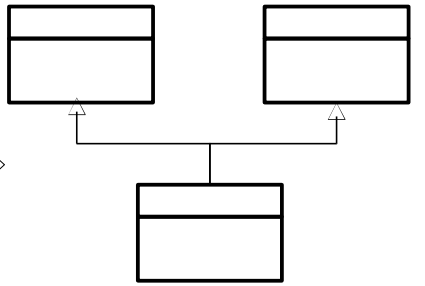
- For software:

```
ADDI $SP, $SP, -8
SW   $A0, 4($SP)
SW   $RA, 0($SP)
```

Machine language

```
For (int i = 0; i < 10; ++i)
    do_something(i);
```

Procedural language

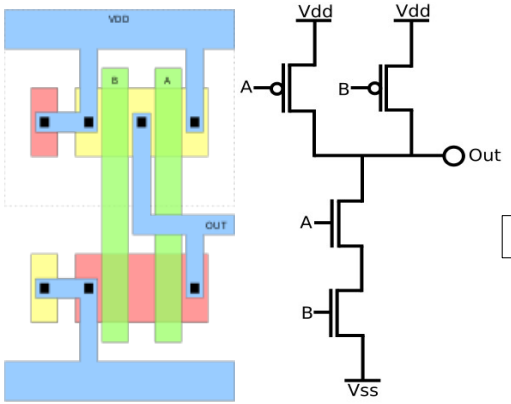


OOP, AOP, UML, etc

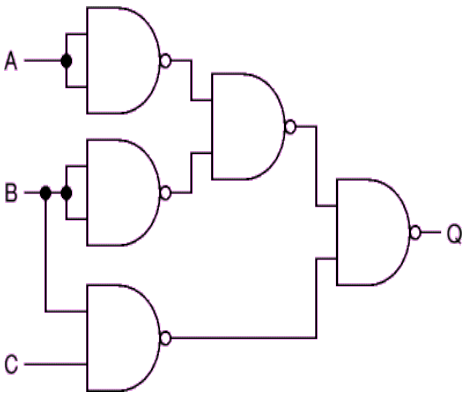
- For hardware:

High-level synthesis

High-level modeling




Electric level




Gate level

```
always @(sel or a or b)
begin
  if (sel == 1)
    f = a;
  else
    f = b;
end
```

Register transfer level

# Evolution of abstraction levels

- For software:

**Convergence between hardware and software design**

```
ADDI $SP, $SP, -8
SW   $A0, 4($SP)
SW   $RA, 0($SP)
```

Machine language

```
For (int i = 0; i < 10; ++i)
    do_something(i);
```

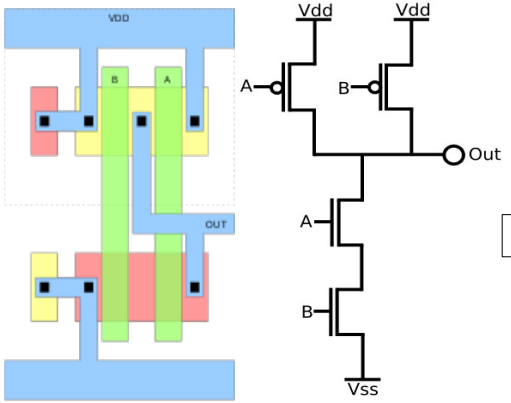
Procedural language

OOP, AOP, UML, etc

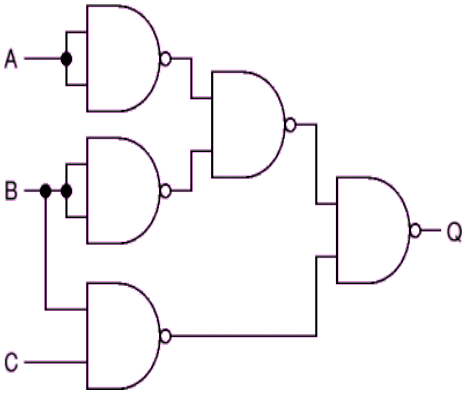
High-level synthesis

High-level modeling

- For hardware:




Electric level




Gate level

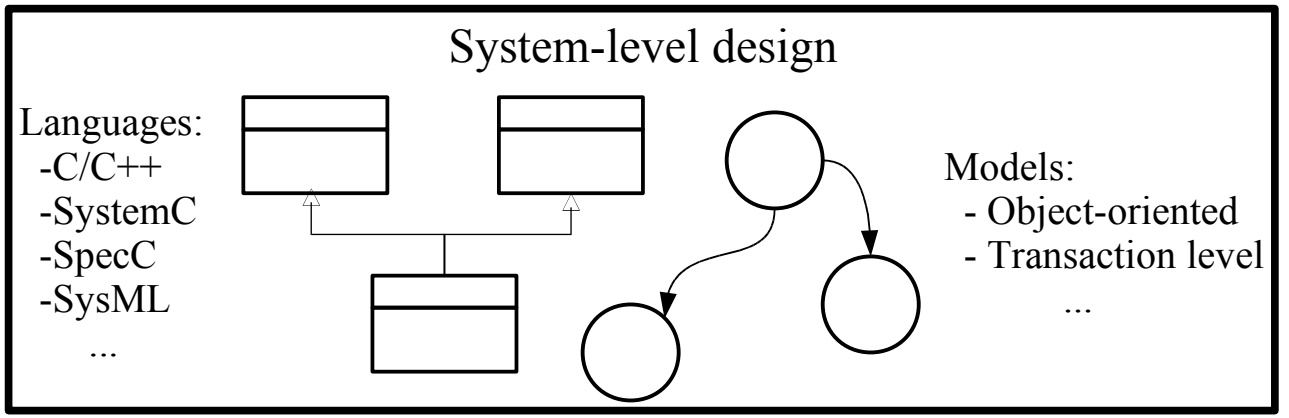
```
always @(sel or a or b)
begin
  if (sel == 1)
    f = a;
  else
    f = b;
end
```

Register transfer level

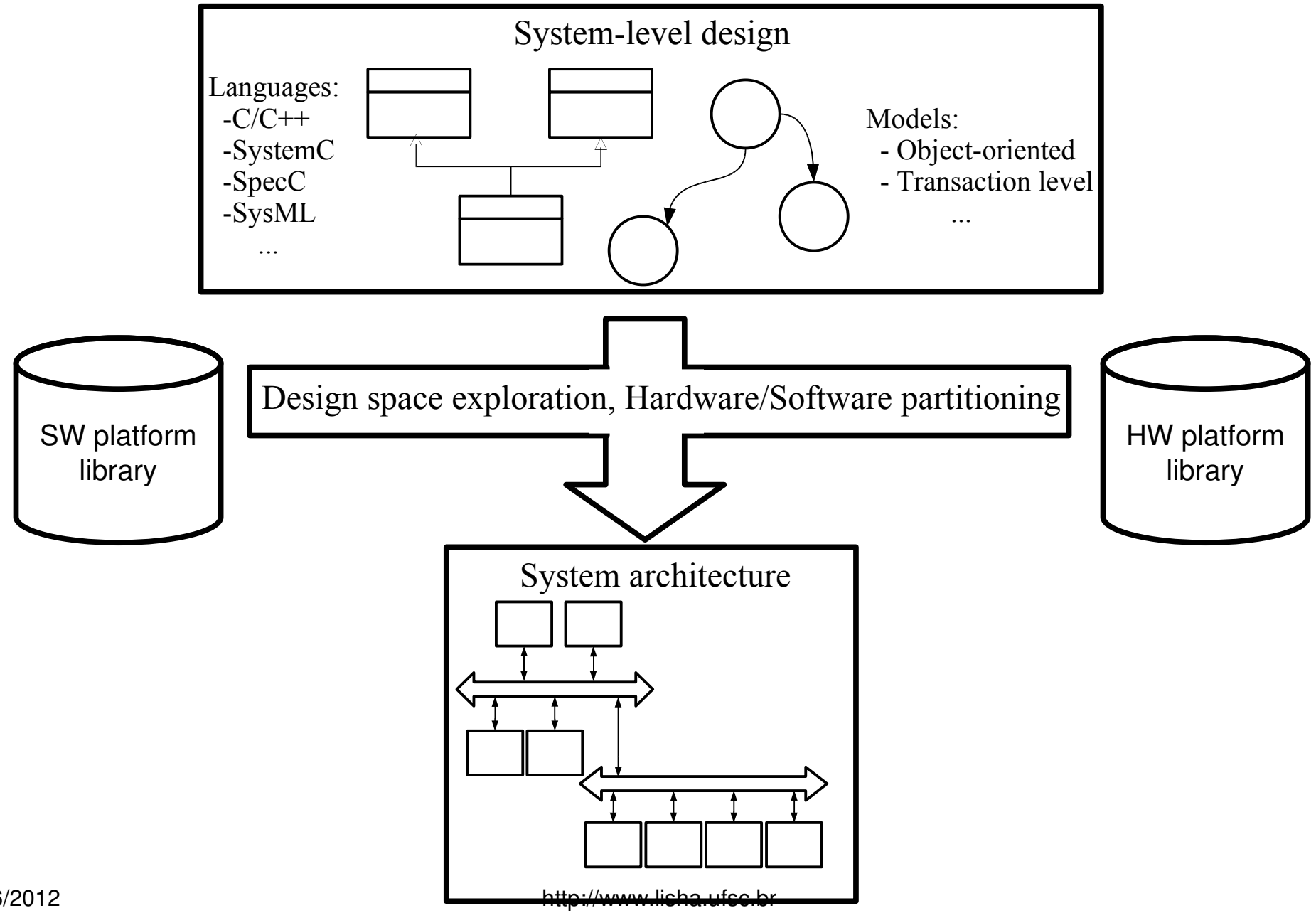
# System-level design


- Use a single high-level language to provide a model of the whole system which is then refined down to the final implementation

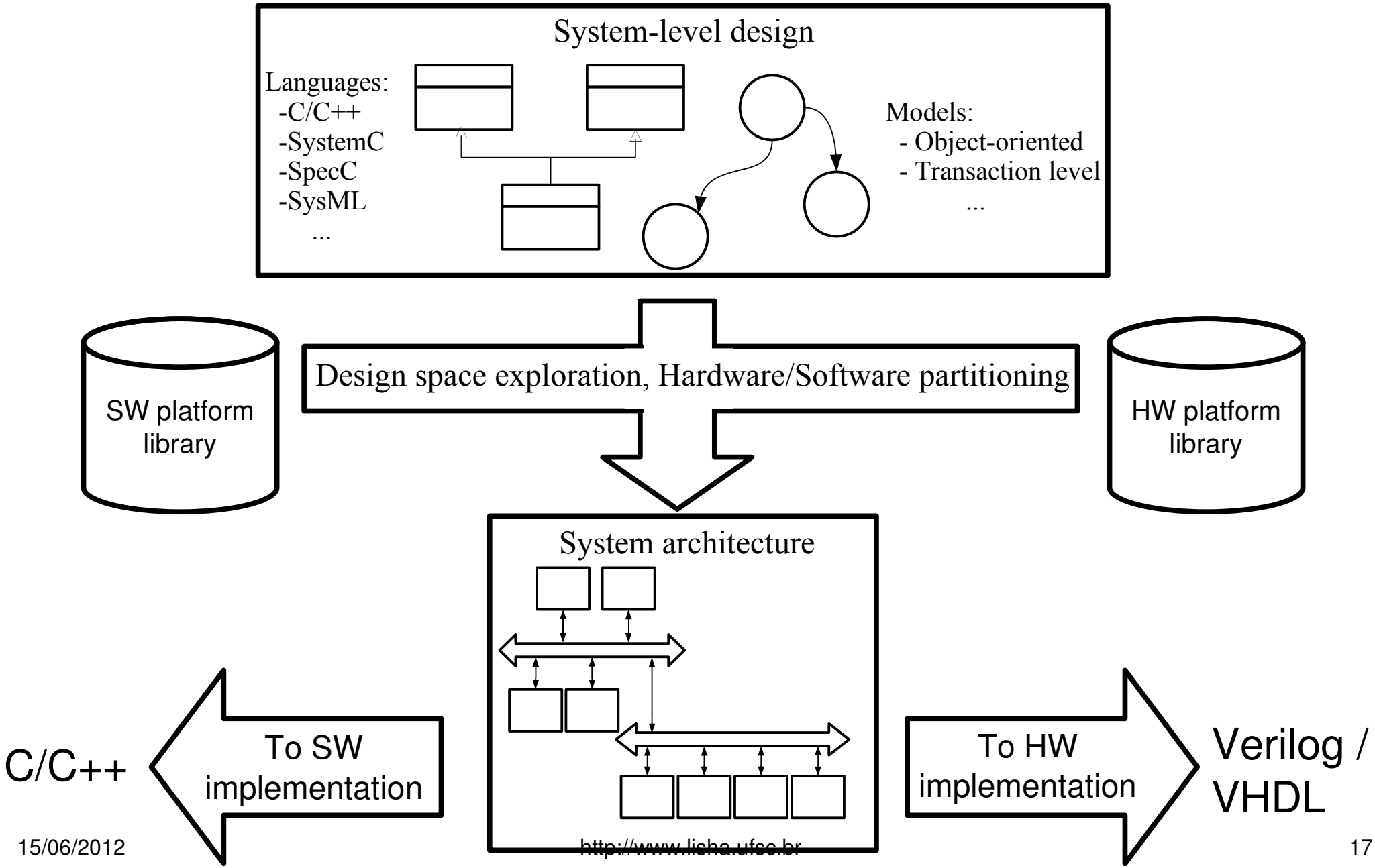
# Today's System-level design flow



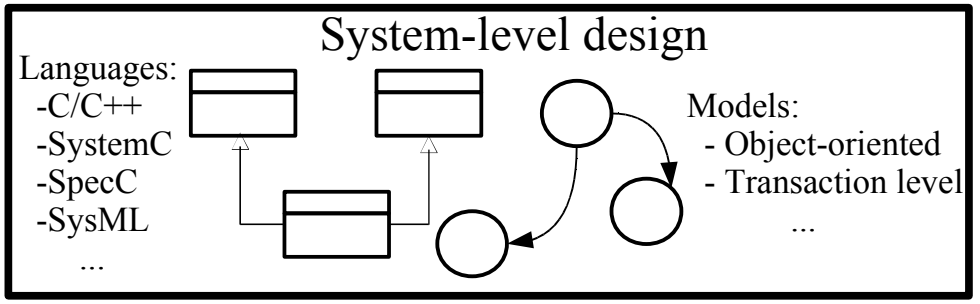
# Today's System-level design flow



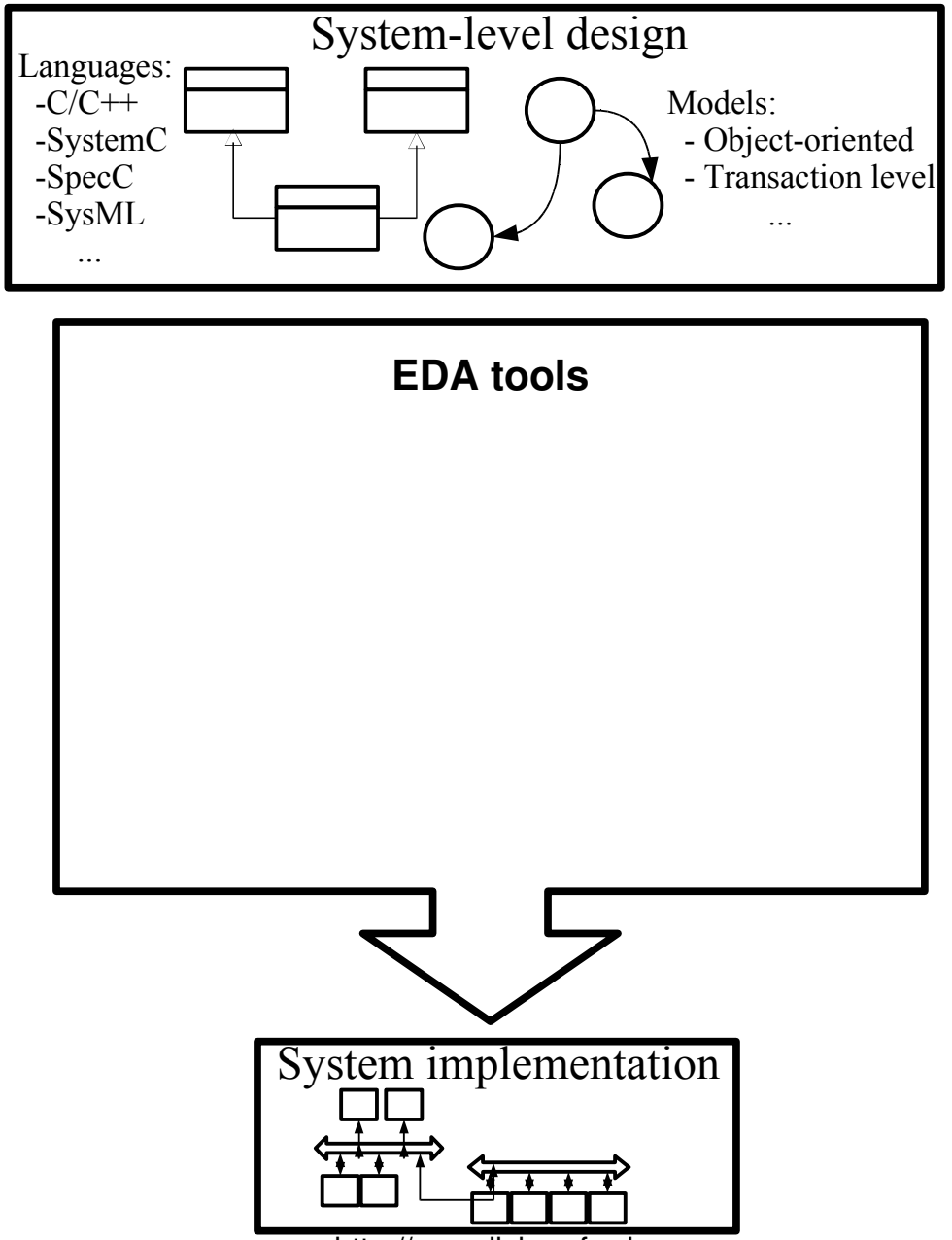
# Today's System-level design flow



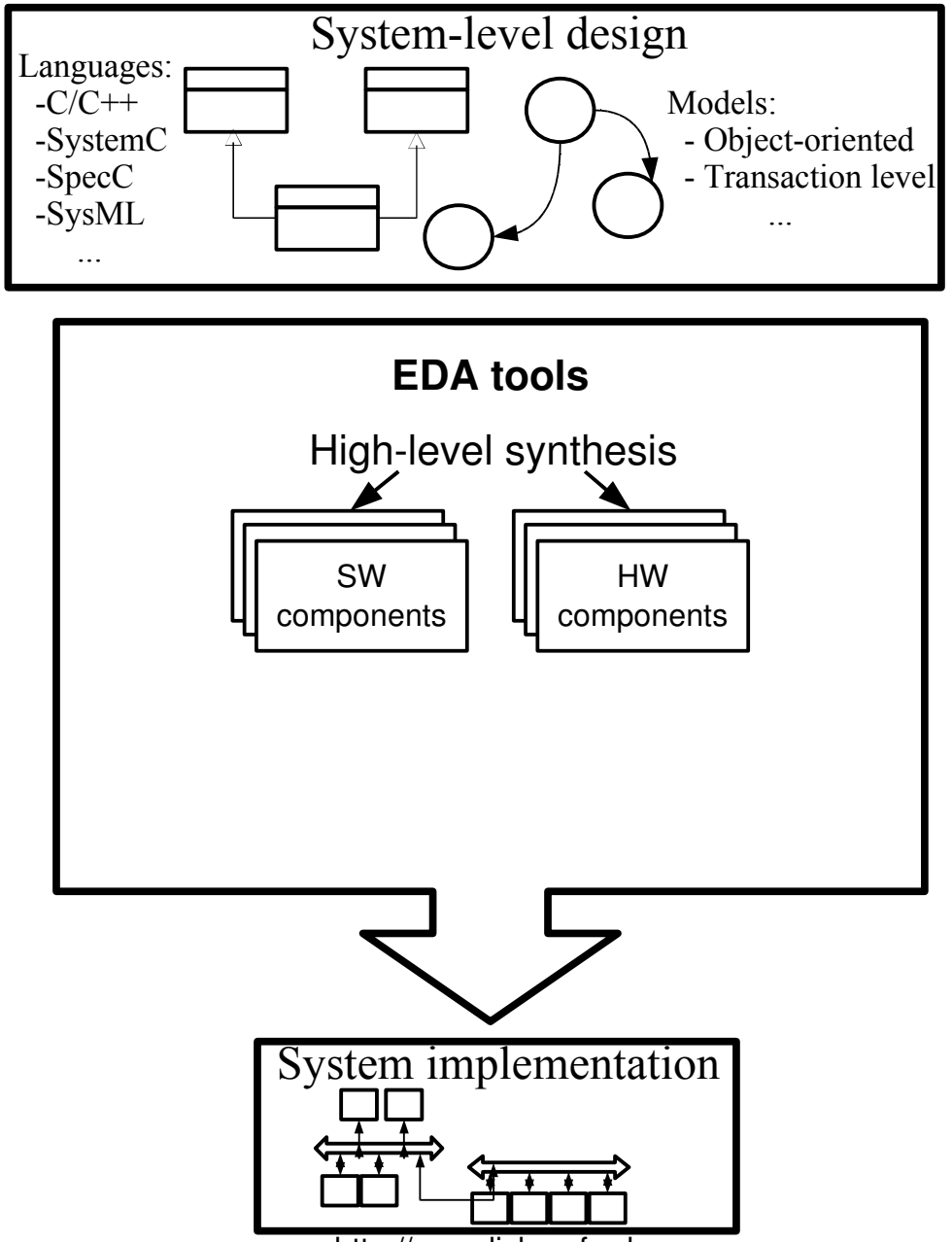
# Tomorrow's System-level design flow



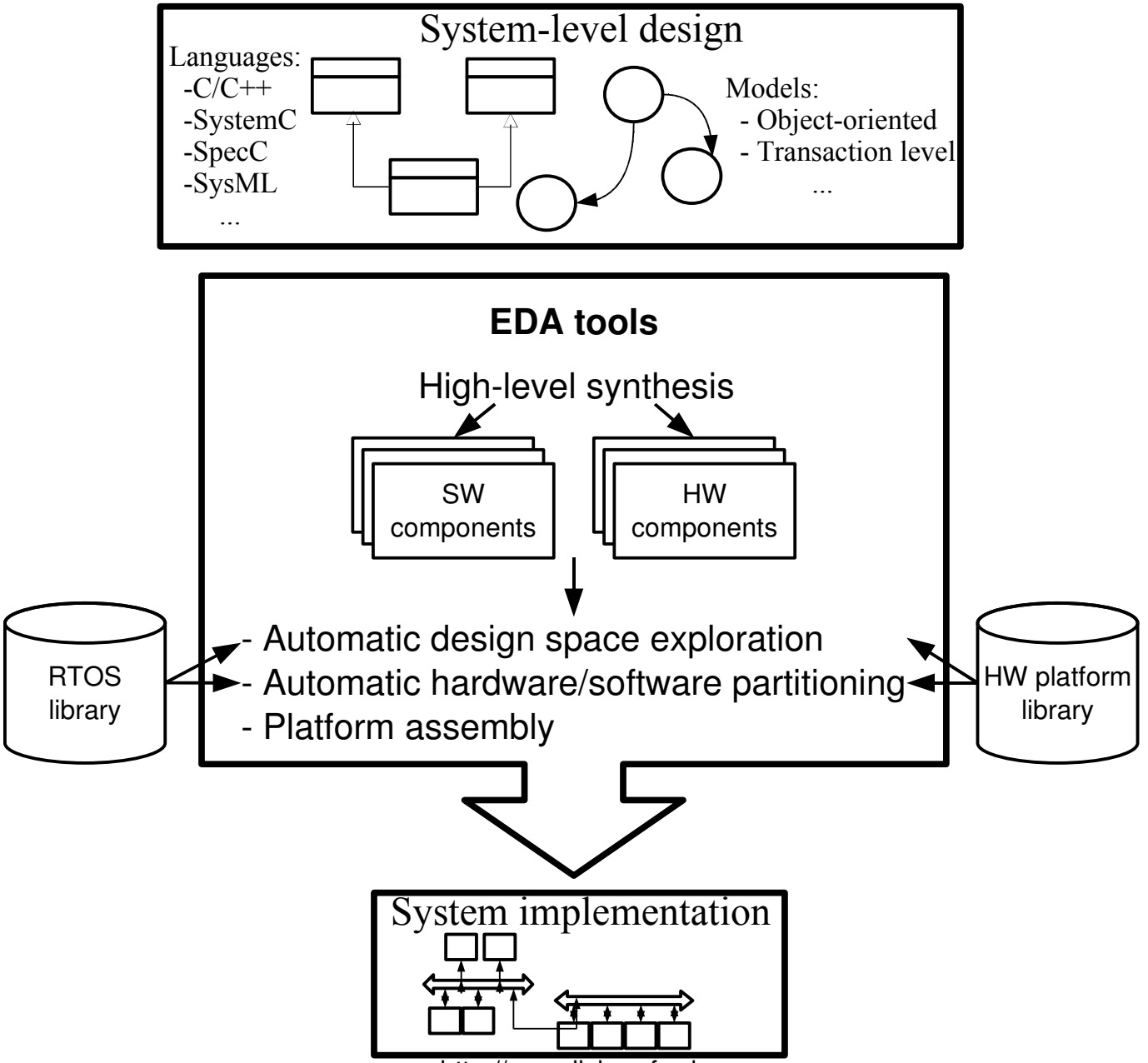
# Tomorrow's System-level design flow



# Tomorrow's System-level design flow

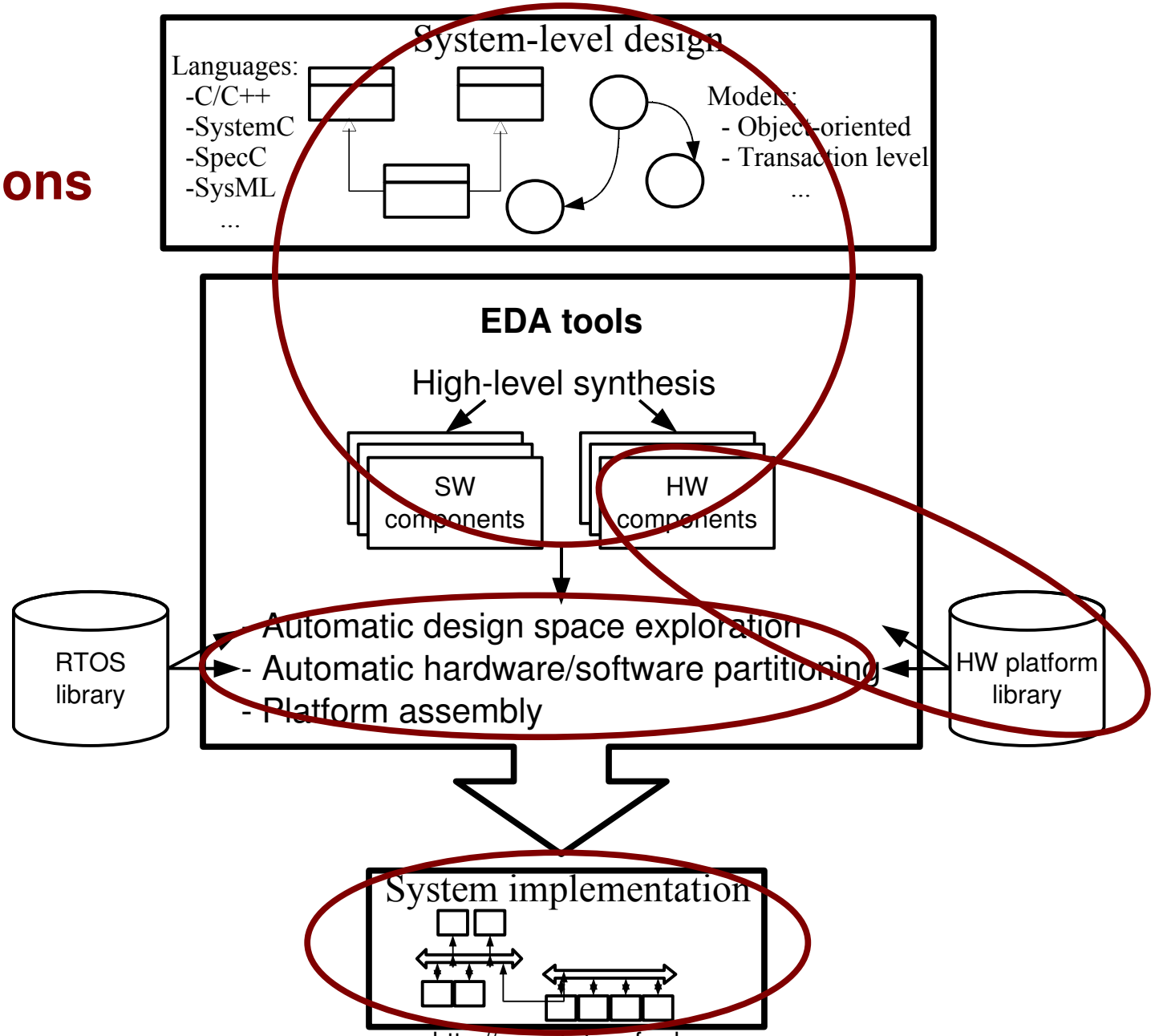


# Tomorrow's System-level design flow



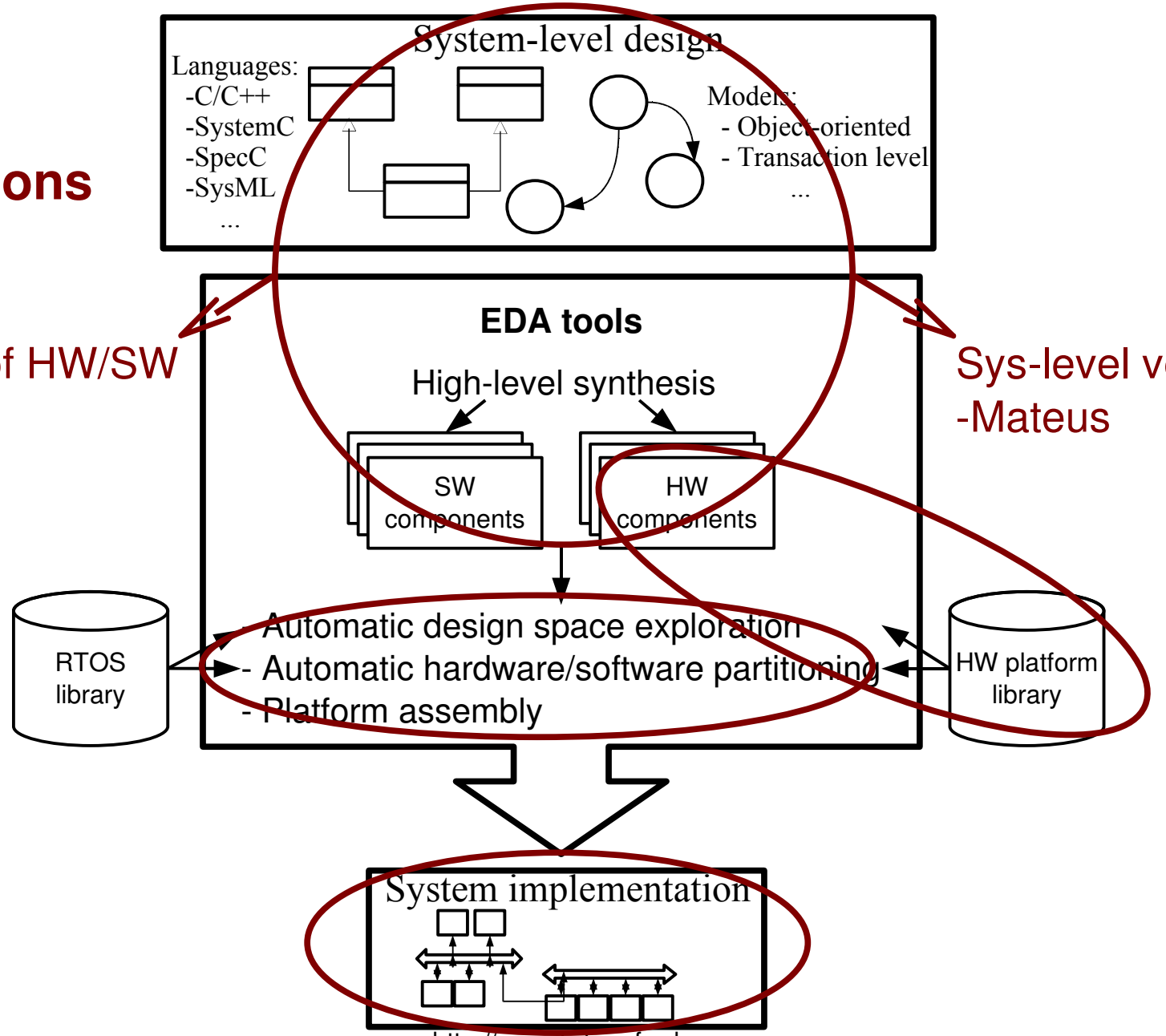
# Tomorrow's System-level design flow

## Our Contributions



# Tomorrow's System-level design flow

## Our Contributions

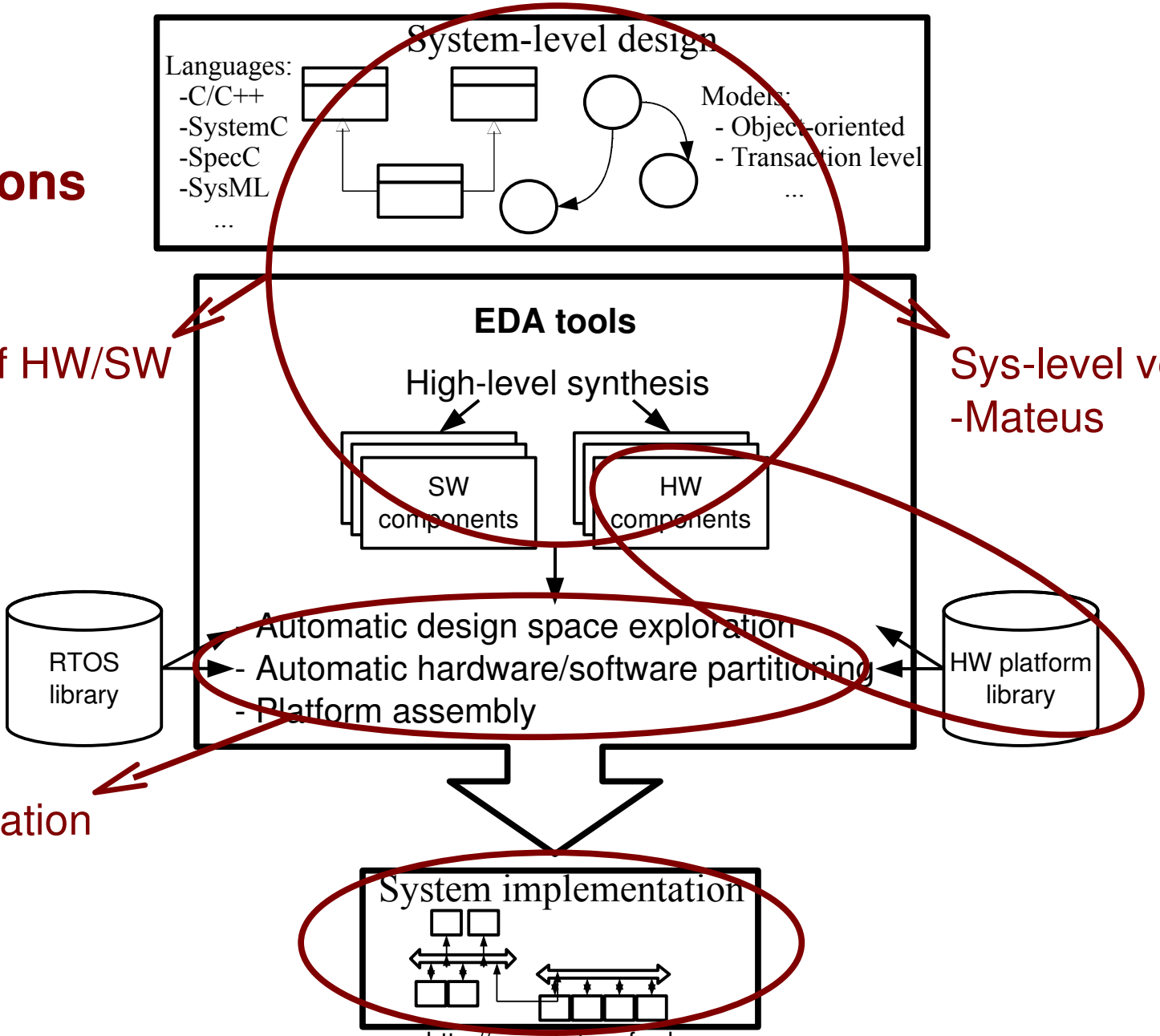



Unif. design of HW/SW
-Tiago

Sys-level verification
-Mateus

# Tomorrow's System-level design flow

## Our Contributions



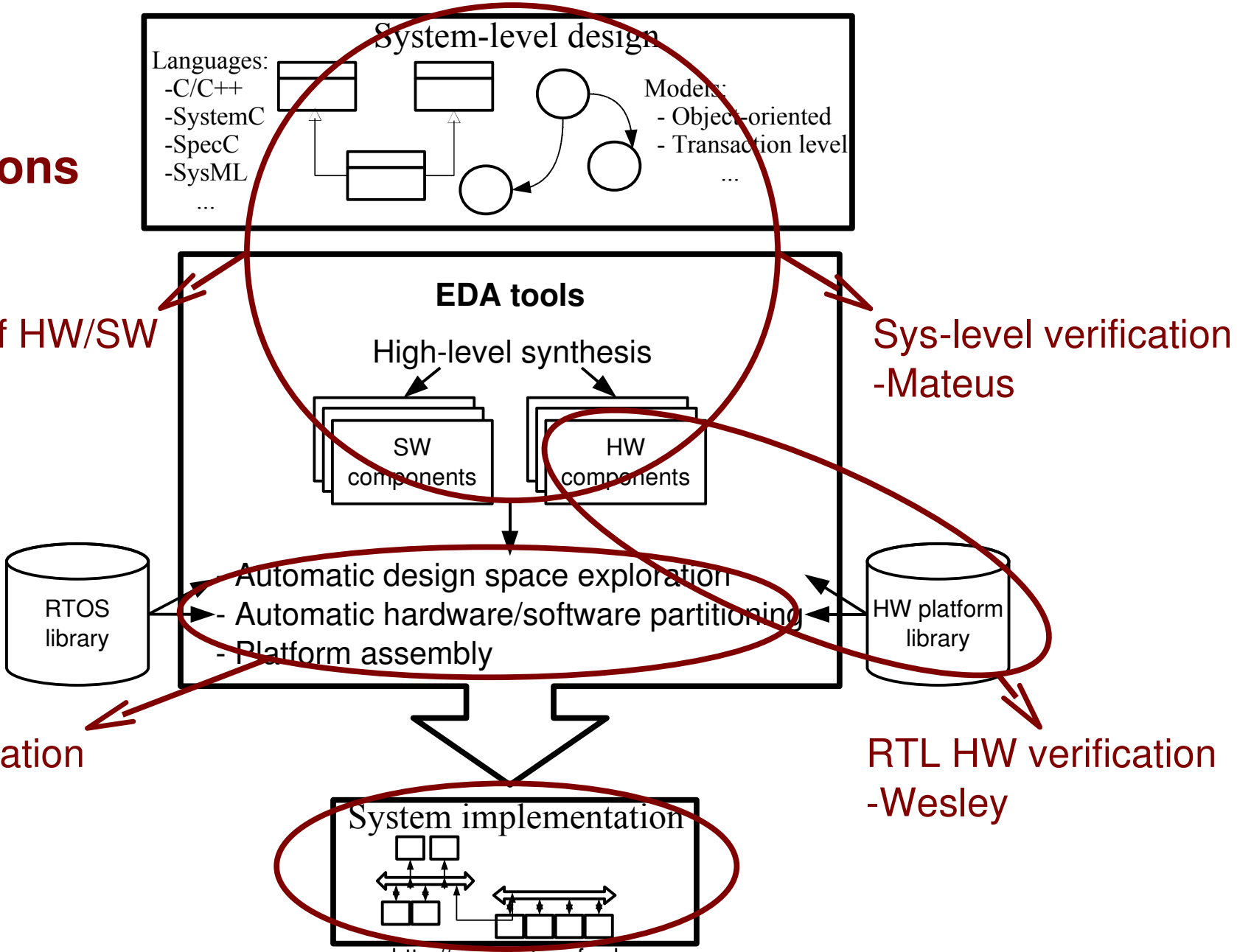

Unif. design of HW/SW
-Tiago

Sys-level verification
-Mateus

ES generation
-Cancian

# Tomorrow's System-level design flow

## Our Contributions



# Tomorrow's System-level design flow

## Our Contributions