Actor-based Runtime Model of Adaptable Feedback Control Loops

Filip Křikava  Philippe Collet  Robert B. France

Université Nice Sophia Antipolis  
I3S - CNRS UMR 7271  
France

Colorado State University  
Computer Science Department  
USA

7th International Workshop on Models@run.time (MRT 2012)  
October, 2nd, 2012 - Innsbruck/AUSTRIA
Capture Feedback Control Loops in networks of reflective actors
Capture Feedback Control Loops in networks of reflective actors

Example: Restart Apache server every time its memory usage exceeds 1GB
Capture Feedback Control Loops in networks of reflective actors

Example: Restart Apache server every time its memory usage exceeds 1GB

- Collecting relevant context information
- Data preprocessing e.g. data filter, rule inference engines, etc.
- Reasoning about system state and planning the appropriate actions to be executed
- Action executing to get a system into a desired state
Capture Feedback Control Loops in networks of reflective actors

**Example:** Restart Apache server every time its memory usage exceeds 1GB

- **Collecting** relevant context information
- Data **preprocessing** e.g. data filter, rule inference engines, etc.
- **Reasoning** about system state and **planning** the appropriate actions to be executed
- Action **executing** to get a system into a desired state
Capture Feedback Control Loops in networks of reflective actors

Example: Restart Apache server every time its memory usage exceeds 1GB

```bash
ps -o pmem=$(cat /var/run/httpd.pid)
```

- Collecting relevant context information
- Data **preprocessing** e.g. data filter, rule inference engines, etc.
- **Reasoning** about system state and **planning** the appropriate actions to be executed
- Action **executing** to get a system into a desired state
Capture Feedback Control Loops in networks of reflective actors

Example: Restart Apache server every time its memory usage exceeds 1GB

- Collecting relevant context information
- Data preprocessing e.g. data filter, rule inference engines, etc.
- Reasoning about system state and planning the appropriate actions to be executed
- Action executing to get a system into a desired state

```
ps -o pmem=$(cat /var/run/httpd.pid)
```

/etc/init.d/apachectl restart
Capture Feedback Control Loops in networks of reflective actors

**Example:** Restart Apache server every time its memory usage exceeds 1GB

```
ps -o pmem=$(cat /var/run/httpd.pid)
```

- **Collecting** relevant context information
- **Data preprocessing** e.g. data filter, rule inference engines, etc.
- **Reasoning** about system state and planning the appropriate actions to be executed
- **Action executing** to get a system into a desired state
Capture Feedback Control Loops in networks of reflective actors

Example: Restart Apache server every time its memory usage exceeds 1GB

```
ps -o pmem=$(cat /var/run/httpd.pid)
```

```
/etc/init.d/apachectl restart
```

Control loop components:
- **Sensor**: Collecting relevant context information
- **Filter**: Data preprocessing e.g. data filter, rule inference engines, etc.
- **Controller**: Reasoning about system state and planning the appropriate actions to be executed
- **Effector**: Action executing to get a system into a desired state
Capture Feedback Control Loops in networks of reflective actors

Example: Restart Apache server every time its memory usage exceeds 1GB

- Collecting relevant context information
- Data preprocessing e.g. data filter, rule inference engines, etc.
- Reasoning about system state and planning the appropriate actions to be executed
- Action executing to get a system into a desired state
Capture Feedback Control Loops in networks of reflective actors

**Example:** Restart Apache server every time its memory usage exceeds 1GB

```bash
ps -o pmem=$(cat /var/run/httpd.pid)
```

```
/etc/init.d/apachectl restart
```

```
ps -o pmem=$(cat /var/run/httpd.pid)
```

```
/etc/init.d/apachectl restart
```

```
ps -o pmem=$(cat /var/run/httpd.pid)
```

```
/etc/init.d/apachectl restart
```

```
ps -o pmem=$(cat /var/run/httpd.pid)
```

```
/etc/init.d/apachectl restart
```

```
ps -o pmem=$(cat /var/run/httpd.pid)
```

```
/etc/init.d/apachectl restart
```
Summary

• **Towards Models@run.time**
  - Alternative form of Models@run.time
  - Explicit and concrete FCLs using reflective actor model
  - Higher-level of abstraction
  - Actors potentially reusable in different scenarios
  - Multiple coordinated and hierarchically layered FCLs
  - Support some non-functional requirements of Models@run.time
    - thread-safety
    - distribution
    - scalability

• **Towards Rapid Prototyping using an MDE toolchain**
  - *Provide researchers and engineers with a tooled approach to experiment with self-adaptive systems*
Thank you!

http://nyx.unice.fr/projects/actress

Filip Křikava
filip.krikava@i3s.unice.fr

The work reported in this paper is partly funded by the ANR SALTY project under contract ANR-09-SEGI-012.