DOL-Critical Tool Chain – Description and Installation

G. Giannopoulou, P. Huang, N. Stoimenov, L. Thiele
Contact: *georgia.giannopoulou@tik.ee.ethz.ch*
ETH Zurich
Computer Engineering and Networks Laboratory
8092 Zurich, Switzerland
Revision April 22, 2014

Contents

This document serves as a short description and installation guide of the software prototypes delivered by ETH Zurich as part of deliverable D7.1 Software Prototypes in the CERTAINTY project. In particular, it describes the DOL-Critical tool chain.

1 DOL-Critical within CERTAINTY

Within the CERTAINTY project, the DOL-Critical tool chain has been developed as part of Work Package 7 to facilitate the (i) specification of mixed-criticality applications, as inspired by the Flight Management System (refer to deliverable D1.1 Criticality and Use Case Requirements) and (ii) the mapping optimization onto architectures resembling one computational cluster of the Kalray MPPA-256. Hence, the target platform for DOL-Critical is Kalray MPPA-256.

The mapping optimization methodology has been developed for the mixed-critical scheduling policy as introduced in Work Package 5: Time-Triggered Scheduling with Synchronization points (see deliverable D5.1 Interference Analysis and Isolation Mechanisms). For the timing analysis of the optimized mapping solution, the scalable interference analysis framework is employed. It is based on modeling the system with timed automata or a combination of timed automata and real-time calculus arrival curves (see deliverable D5.1 Interference Analysis and Isolation Mechanisms).

2 Related Publications

The conference publications describing the analysis models and methods utilized in the DOL-Critical tool chain in terms of mapping optimization and timing / interference analysis are summarized below:

- G. Giannopoulou, N. Stoimenov, P. Huang and L. Thiele. *Mapping Mixed-Criticality Applications on Multi-Core Architectures.* In Proc. Design, Automation & Test in Europe (DATE), Hot-Topic Session on Predictable Multicore Computing, Dresden, Germany, March 2014.

- G. Giannopoulou, N. Stoimenov, P. Huang and L. Thiele. *Scheduling of Mixed-Criticality Applications on Resource-Sharing Multicore Systems.* In Proc. International Conference on Embedded Software (EMSOFT), p. 17:1-17:15, Montreal, Canada, October 2013.

- G. Giannopoulou, K. Lampka, N. Stoimenov and L. Thiele. *Timed Model Checking with Abstractions: Towards Worst-Case Response Time Analysis in Resource-Sharing Manycore Systems.* In Proc. International Conference on Embedded Software (EMSOFT), p. 63-72, Tampere, Finland, October 2012.

3 The DOL-Critical Flow - Overview

The document continues with a description of the tool flow in DOL-Critical.

1. A mixed-criticality application specification which is a combination of XML format description of application tasks (described in deliverable D2.3 Modelling Languages and Models) and the C code of the tasks (written in the C programming style defined in deliverable D6.1 Formal Composition Language and Validation Research Analysis) is parsed. The XML application specification already contains the results from the static timing analysis (as developed in Work Package 3) for each task for the considered hardware platform in terms of execution times and resource accesses per superblock phase as they are needed for optimizing the mapping.

2. An XML format specification of the hardware platform architecture (described in deliverable D2.3 Modelling Languages and Models) is parsed.

3. An executable Linux application running on a single processor is generated for the purpose of functional simulation of the application.

4. An optimized mapping of the application onto the hardware platform is generated. It is in XML format (described in deliverable D2.3 Modelling Languages and Models) and includes the scheduling parameters for all used processing cores. The considered scheduling policy is the Time-Triggered Scheduling with Synchronization points (as described in deliverable D5.1 Interference Analysis and Isolation Mechanisms).

5. The generated mapping together with the application and platform specifications are used by the BIP framework to generate a BIP system model which is used for formal verification and code generation. However, this is not part of DOL-Critical and not considered in this document.

Fig. 1 gives an overview of the tool flow in DOL-Critical. You may also visit http://www.tik.ee.ethz.ch/∼certainty for further information on the DOL-Critical framework.

4 The DOL-Critical Tool Chain

4.1 Input Description

The application programmer has to provide the following XML files:

- an XML file according to the application XML Schema definition `schema/DOLC.xsd` which describes the application, i.e., the mixed-

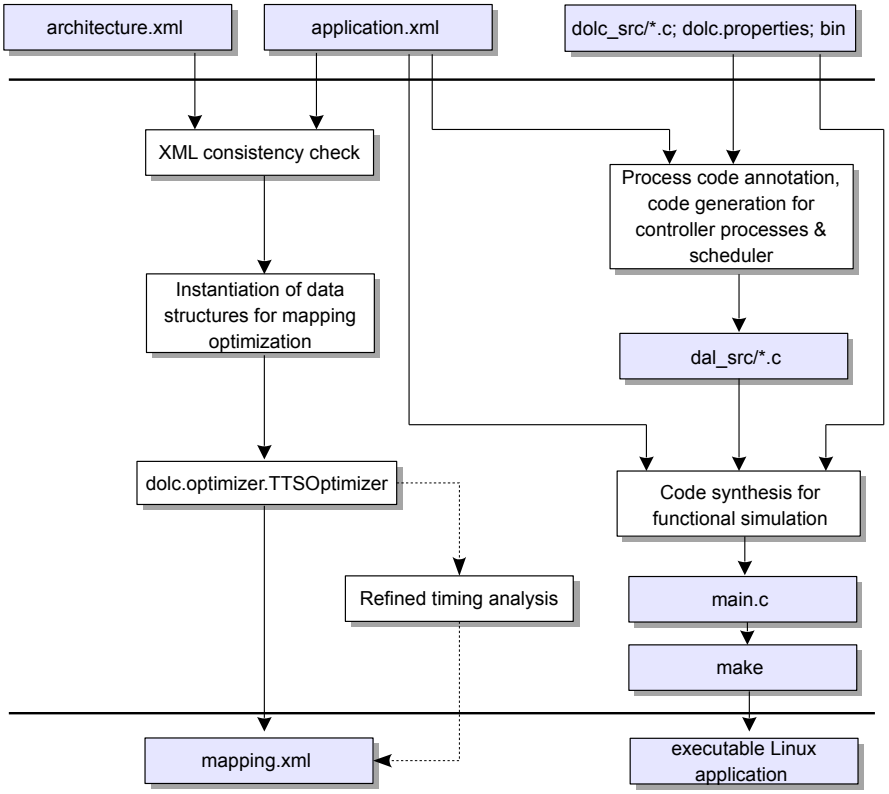

Figure 1: Tool chain.

criticality processes, their controllers, control dependencies, communication channels, and precedence constraints,

- an XML file according to the architecture XML Schema definition `schema/architecture.xsd` which describes the target multi-core shared-memory platform.

The XML specifications of the application and the platform contain all necessary information for mapping optimization. Furthermore, the application programmer has to provide a C source code file for each process. With the tool chain described in this section, these sources can be converted into an executable Linux application that runs on a single processor. This enables the functional simulation of the application so that the programmer can identify inconsistencies or logical bugs early in the design phase.

4.2 Dependencies and Requirements

The main tool of the tool chain, **dolc**, is written in Java. Therefore, a Java compiler **javac** and the Java application launcher **java** are used in the tool

chain. In the code, Java features are used that have been newly introduced in the Java Platform 7.0. Therefore, appropriate versions of **javac** and **java** are required (**javac** and **java** version 1.7 as included in the J2SE 7.0 JDK). For creating the Linux application, the C/C++ compiler **g++** (version 4.3 or higher) is used. Besides these standard tools, the following libraries and tools are used in the tool chain:
As libraries,

- **jdom:** http://www.jdom.org/ (version 1.0),
- **xerces:** http://xerces.apache.org/xerces2-j/ (version 2.8.0),
- **commons-lang3:** http://commons.apache.org/proper/commons-lang/ (version 3.2.1)

Note that the **jdom**, **xerces** and **commons-lang3** libraries are already contained in the DOLC distribution (directory `bin/`), so there is no need to download them.
As build tools,

- **ant:** http://ant.apache.org/ (version 1.7.0),
- **ant-contrib:** http://sourceforge.net/projects/ant-contrib (version 1.0b3), and
- **make:** http://www.gnu.org/software/make/ (version 3.81) are used.

The tool chain has been tested only on 64-bit systems. Therefore, compatibility with 32-bit systems is *not* guaranteed.

4.3 Main Tools

The following paragraphs describe the main tasks of the tool chain, namely the functional simulation and the mapping optimization. For each task, a series of steps has to be followed, as shown in Fig. 1. Note that the archive file `dolc_ethz.zip` contains an **ant** build file, which performs these steps in an automated manner.

The first task in the DOL-Critical tool chain, which is executed through steps `prepareDolc` to `execute` in the **ant** file `examples/runexample.xml`, is the **functional simulation** of the input application. The application is specified in XML (definition of processes, controllers, data and control channels, precedence constraints) and C code (process functionality) according to the DOL-Critical API. For this task, we have implemented a front-end to an existing simulation environment, which was developed within the Distributed Application Layer (DAL) framework (http://www.tik.ee.ethz.ch/∼euretile/dal.php). This interface between the DOL-Critical application specification and the DAL simulation environment features the following operations:

- Synthesize code for the process controllers (not given by the application programmer) based on the XML specification.
- Synthesize code for a scheduler that simply triggers execution of the processes based on their activation patterns on a single core. The scheduler maintains a timestamp queue with the timestamps of the processes' forthcoming activations. This queue is updated by interaction with the process controllers.

After these two steps are performed, the `dal_src/` directory is generated under `dolc_ethz/build/bin/main/exampleDemo/`. This directory contains the C source code for all processes, their controllers and the single-core scheduler, which can be given as input together with the original XML specification to the DAL simulation environment. The next step is:

- Generate process wrappers for execution of all above processes on a single processor under Linux.

The outcome of the last step is an executable Linux application.

Note that during functional simulation, the process criticality levels, execution times, and precedence constraints are neglected as the objective is to identify inconsistencies with respect to the DOL-Critical API and / or logical mistakes in the application specification. Nevertheless, the process activation patterns are respected, since this enables us to test the inter-process communication patterns and examine for instance, if the order of writing data to specific communication channels is deterministic, if mailboxes with greater capacity are needed, if certain written data are never read, etc.

When executing the auto-generated runnable application, the following information is printed out:

- For each process controller: the time points when it triggers execution of its controlled process (after having been triggered itself by the scheduler).
- For each process controller: the timestamps that it transmits to the scheduler concerning the next activation of its controlled process.
- For each process: the data that it read/wrote from/to data communication channels, if the respective print messages have been enabled by the application programmer (recommended, see e.g., coding style in `examples/exampleDemo/dolc-src/` C source code).

If one needs to de-activate functional simulation from the automated tool chain, the operations `prepareDolc` to `execute` in the target `runexample` of the **ant** file `examples/runexample.xml` have to be commented out.

The second task in the DOL-Critical tool chain, which is executed through step `mapping-optimization` in the **ant** file `examples/runexample.xml`, is the **mapping optimization** of the XML-specified application onto the multicore architecture, which is also given as XML input. This task triggers the following steps:

- The application and architecture XML files go through consistency check with respect to the corresponding XML schemata semantics.
- The required information for mapping optimization (class `dolc.optimizer.Specification`) are collected and given as input to the mapping optimizer.
- The optimizer (class `dolc.optimizer.TTSOptimizer`) performs design space exploration with a time budget of 1 hour. Design space exploration stops when it converges to one mapping solution or the time budget is exhausted. The mapping optimization is performed under the Time-Triggered scheduling policy with Synchronization points (TTS) which has been introduced in the CERTAINTY project deliverable D5.1 'Interference Analysis and Isolation Mechanisms'. For the estimation of the worst-case lengths of the TTS sub-frames, the timing analysis for each visited mapping solution accounts for the effects of memory contention (conservative analysis). The objective function of the optimization problem is defined as the 3rd norm (L_3) of the estimated sub-frame lengths across all TTS frames and all criticality scenarios (assumed worst-case task execution times). Based on this, the optimal mapping solution is the one with minimal TTS sub-frame lengths, which implies a balanced workload among the cores (parallelism is fully exploited), increased schedulability, and maximal "spare" time for the scheduling of future tasks, if such ones exist.

The outcome of this task is a TTS mapping specification, i.e., a binding of tasks to cores, schedule tables for each core, and the estimated TTS sub-frame lengths under different criticality scenarios, which is saved in XML format.

Note that for the best found solution, which is saved in `mapping.xml`, timing analysis can be repeated using more accurate analysis methods, such as the timed automata based approach, which was presented in the CERTAINTY project deliverable D5.1 'Interference Analysis and Isolation Mechanisms'.

5 Automated Operation of the DOL-Critical Tool Chain

The DOL-Critical package comes with a demo example for which the tool chain can be run automatically. Before doing so, make sure that the follow-

ing tools are in place (refer to the previous section for links to obtain those tools):

1. C/C++ environment: compiler, linker (g++ version 4.3 or greater)
2. Java environment: javac, java (version 1.7)
3. Build environment: make, Ant (version 1.7.0 or greater), ant-contrib

To run the DOL-Critical tool chain on a **64-bit** system, simply walk through the following step-by-step instructions. The given procedure has been tested under Ubuntu Linux 12.04, Debian Linux 7, and Windows 8 with Cygwin 1.7.23 (all 64-bit versions).

1. Copy the `dolc_ethz.zip` archive to some directory, for instance `dolc_ethz`:

   ```
   ~/dolc_ethz>unzip dolc_ethz.zip
   ~/dolc_ethz>cd dolc_ethz
   ```

2. After unzipping the archive, a file `build_zip.xml` will be located in this directory (**~/dolc_ethz>**). Use **ant** to generate the `src/dolc.properties` file based on the settings and copy it to the `dolc.jar`:

   ```
   ~/dolc_ethz>ant -f build_zip.xml config
   ```

3. Use **ant** to set up the build directory structure `build/bin/main` and copy all the necessary resources to this directory:

   ```
   ~/dolc_ethz>ant -f build_zip.xml compile
   ```

(3a.) Note: You may also use the `all` target to perform the previous two steps in one call:

```
~/dolc_ethz>ant -f build_zip.xml all
```

4. Change to `build/bin/main` and run the example using the provided **ant** build file `runexample.xml`. If not modified, `runexample.xml` will trigger the functional simulation and the mapping optimization of the specified example. For the mapping optimization, it is necessary to provide the TTS dimensions, i.e., the scheduling cycle period and the frame lengths. For the demonstrated example, we take them equal to 200 ms and 50 ms, respectively. We also assume that the application and architecture specifications are given in `dolc_ethz/examples/exampleDemo/exampleDemo-DOLC.xml` and `dolc_ethz/examples/arch/archDemo.xml`, respectively.

   ```
   ~/dolc_ethz>cd build/bin/main
   ~/dolc_ethz/build/bin/main>ant -f runexample.xml
   -Dnumber=Demo -Dplatformfile="archDemo.xml"
   -Dschedcycle=0.2 -Dframelength=0.05
   ```

An example output for **ant** is shown below.

```
Buildfile: /home/user/dolc_ethz/build/bin/main/runexample.xml

showversion:

showantversion:
[echo] Use Apache Ant(TM) version 1.8.2 compiled on December 3 2011.

showjavaversion:
[echo] Use Java version 1.7.0_51 (required version: 1.7.0 or higher)
    .

showjavacversion:
[echo] Use Java version 1.7.0_51 (required version: 1.7.0 or higher)
    .

runexample:

prepareDolc:
[echo] Create directory exampleDemo.
[echo] Create directory /home/user/dolc_ethz/build/bin/main/
    exampleDemo.
[echo] Copy C source files.
[mkdir] Created dir: /home/user/dolc_ethz/build/bin/main/exampleDemo
    /dolc-src
[copy] Copying 16 files to /home/user/dolc_ethz/build/bin/main/
    exampleDemo/dolc-src
[copy] Copying 1 file to /home/user/dolc_ethz/build/bin/main/
    exampleDemo
[mkdir] Created dir: /home/user/dolc_ethz/build/bin/main/exampleDemo
    /dal-src/app1/src

dolc:
[copy] Copying 6 files to /home/user/dolc_ethz/build/bin/main/
    exampleDemo/dal-src/app1/src
[copy] Copying 1 file to /home/user/dolc_ethz/build/bin/main/
    exampleDemo/dal-src

flattenFsm:
[copy] Copying 1 file to /home/user/dolc_ethz/build/bin/main/
    exampleDemo/dal-src

flattenPN:
[echo] Create flattened XML /home/user/dolc_ethz/build/bin/main/
    exampleDemo/dal-src/app1/pn_flattened.xml.
[java]
    ..........................................................
[javac] Compiling 1 source file to /home/user/dolc_ethz/build/bin/
    main/exampleDemo/dal-src/app1

dalPreparation:
[echo] Run DAL Preparation.
[java] Read finite state machine from XML file
[java] -- full filename: file:/home/user/dolc_ethz/build/bin/main/
    exampleDemo/dal-src/exampleDemo_flattened.xml
[java] Read process network from XML file
[java] -- full filename: file:/home/user/dolc_ethz/build/bin/main/
    exampleDemo/dal-src/app1/pn_flattened.xml
[java] -- Process network model from XML [Finished]
[java]
[java] -- Finite state machine model from XML [Finished]
[java]
```

```
47 [java] Read architecture from XML file
48 [java] -- full filename: file:/home/user/dolc_ethz/examples/arch/
       archDemo.xml
49 [java] -- Architecture model from XML [Finished]
50 [java]
51 [java] Generating Preparation package:
52 [java] -- Consistency check:
53 [java] APPL: Checking resource name ...
54 [java] APPL: Checking channel ports ...
55 [java] APPL: Checking channel connection ...
56 [java] APPL: Checking Process connection ...
57 [java] ARCH: Checking resource name ...
58 [java] ARCH: Checking network simulators ...
59 [java] -- Consistency check [Finished]
60 [java]
61 [java] -- Generation [Finished]
62 [java]
63
64 dalFunctionalController:
65 [echo] Run Functional Controller Synthesis.
66 [java] Read finite state machine from XML file
67 [java] -- full filename: file:/home/user/dolc_ethz/build/bin/main/
       exampleDemo/dal-src/exampleDemo_flattened.xml
68 [java] Read process network from XML file
69 [java] -- full filename: file:/home/user/dolc_ethz/build/bin/main/
       exampleDemo/dal-src/app1/pn_flattened.xml
70 [java] -- Process network model from XML [Finished]
71 [java]
72 [java] -- Finite state machine model from XML [Finished]
73 [java]
74 [java] Generating Functional Controller package:
75 [java] -- Generation [Finished]
76 [java]
77
78 dal:
79 [echo] Run DAL.
80 [java] Read architecture from XML file
81 [java] -- full filename: file:/home/user/dolc_ethz/build/bin/main/
       exampleDemo/dal-src/controller/platform.xml
82 [java] -- Architecture model from XML [Finished]
83 [java]
84 [java] Read process network from XML file
85 [java] -- full filename: file:/home/user/dolc_ethz/build/bin/main/
       exampleDemo/dal-src/controller/APP1/processnetwork.xml
86 [java] -- Process network model from XML [Finished]
87 [java]
88 [java] Read process network from XML file
89 [java] -- full filename: file:/home/user/dolc_ethz/build/bin/main/
       exampleDemo/dal-src/controller/Controller/processnetwork.xml
90 [java] -- Process network model from XML [Finished]
91 [java]
92 [java] Read mapping from XML file
93 [java] -- full filename: file:/home/user/dolc_ethz/build/bin/main/
       exampleDemo/dal-src/controller/mapping.xml
94 [java] -- Mapping from XML [Finished]
95 [java]
96 [java] Generating SingleLinux-package:
97 [java] Created directory singlelinux/tmp/util
98 [java] Created directory singlelinux/tmp/include
99 [java] Created directory singlelinux/tmp/util
100 [java] Created directory singlelinux/tmp/include
101 [java] -- Generation [Finished]
```

```
[java]

execute:
[echo] Make singlelinux application.
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/channel.cpp -o
    util/channel.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/checutil/channel
    .cpp -o util/checkpoint.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/
    controllerHelperGeneric.cpp -o util/controllerHelperGeneric.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/
    controllerHelperSpecific.cpp -o util/controllerHelperSpecific.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/faultSimulator.
    cpp -o util/faultSimulator.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/Fifo.cpp -o util
    /Fifo.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/
    InnerProcessorFifo.cpp -o util/InnerProcessorFifo.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/
    setupHelperGeneric.cpp -o util/setupHelperGeneric.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/
    setupHelperSpecific.cpp -o util/setupHelperSpecific.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/sleep.cpp -o
    util/sleep.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/Timer.cpp -o
    util/Timer.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude util/videoHelper.cpp
    -o util/videoHelper.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -Iinclude main.cpp -o main.o
[exec] g++ -pthread -lm *.o util*.o -o main -ldl
[exec] cd app/APP1; make DAL_TESTING=
[exec] make[1]: Entering directory `/home/user/dolc_ethz/build/bin/
    main/exampleDemo/dal-src/singlelinux/app/APP1'
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include dolc.c -o
    dolc.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include periodic1.c
     -o periodic1.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include periodic2.c
     -o periodic2.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include periodic3.c
     -o periodic3.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include periodic4.c
     -o periodic4.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include periodic5.c
     -o periodic5.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include periodic6.c
     -o periodic6.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include dolcsched/
    dolcsched.c -o dolcsched/dolcsched.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include dolcsched/
    wrap_process.c -o dolcsched/wrap_process.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
    periodic1Cntr/periodic1Cntr.c -o periodic1Cntr/periodic1Cntr.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
    periodic1Cntr/wrap_process.c -o periodic1Cntr/wrap_process.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
    periodic2Cntr/periodic2Cntr.c -o periodic2Cntr/periodic2Cntr.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
    periodic2Cntr/wrap_process.c -o periodic2Cntr/wrap_process.o
[exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
    periodic3Cntr/periodic3Cntr.c -o periodic3Cntr/periodic3Cntr.o
```

```
136 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        periodic3Cntr/wrap_process.c -o periodic3Cntr/wrap_process.o
137 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        periodic4Cntr/periodic4Cntr.c -o periodic4Cntr/periodic4Cntr.o
138 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        periodic4Cntr/wrap_process.c -o periodic4Cntr/wrap_process.o
139 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        periodic5Cntr/periodic5Cntr.c -o periodic5Cntr/periodic5Cntr.o
140 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        periodic5Cntr/wrap_process.c -o periodic5Cntr/wrap_process.o
141 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        periodic6Cntr/periodic6Cntr.c -o periodic6Cntr/periodic6Cntr.o
142 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        periodic6Cntr/wrap_process.c -o periodic6Cntr/wrap_process.o
143 [exec] g++ -pthread -shared -Wl,-soname,libperiodic6Cntr.so -o
        libperiodic6Cntr.so ../../util/*.o periodic6Cntr/*.o *.o
144 [exec] mv libperiodic6Cntr.so "../../lib/APP1/libperiodic6Cntr.so"
145 [exec] g++ -pthread -shared -Wl,-soname,libperiodic5Cntr.so -o
        libperiodic5Cntr.so ../../util/*.o periodic5Cntr/*.o *.o
146 [exec] mv libperiodic5Cntr.so "../../lib/APP1/libperiodic5Cntr.so"
147 [exec] g++ -pthread -shared -Wl,-soname,libperiodic3Cntr.so -o
        libperiodic3Cntr.so ../../util/*.o periodic3Cntr/*.o *.o
148 [exec] mv libperiodic3Cntr.so "../../lib/APP1/libperiodic3Cntr.so"
149 [exec] g++ -pthread -shared -Wl,-soname,libdolcsched.so -o
        libdolcsched.so ../../util/*.o dolcsched/*.o *.o
150 [exec] mv libdolcsched.so "../../lib/APP1/libdolcsched.so"
151 [exec] g++ -pthread -shared -Wl,-soname,libperiodic4Cntr.so -o
        libperiodic4Cntr.so ../../util/*.o periodic4Cntr/*.o *.o
152 [exec] mv libperiodic4Cntr.so "../../lib/APP1/libperiodic4Cntr.so"
153 [exec] g++ -pthread -shared -Wl,-soname,libperiodic1Cntr.so -o
        libperiodic1Cntr.so ../../util/*.o periodic1Cntr/*.o *.o
154 [exec] mv libperiodic1Cntr.so "../../lib/APP1/libperiodic1Cntr.so"
155 [exec] g++ -pthread -shared -Wl,-soname,libperiodic2Cntr.so -o
        libperiodic2Cntr.so ../../util/*.o periodic2Cntr/*.o *.o
156 [exec] mv libperiodic2Cntr.so "../../lib/APP1/libperiodic2Cntr.so"
157 [exec] make[1]: Leaving directory `/home/user/dolc_ethz/build/bin/
        main/exampleDemo/dal-src/singlelinux/app/APP1'
158 [exec] cd app/Controller; make DAL_TESTING=
159 [exec] make[1]: Entering directory `/home/user/dolc_ethz/build/bin/
        main/exampleDemo/dal-src/singlelinux/app/Controller'
160 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        APP1_machine.c -o APP1_machine.o
161 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include
        masterhelper.c -o masterhelper.o
162 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include message.c -
        o message.o
163 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include master/
        master.c -o master/master.o
164 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include master/
        wrap_process.c -o master/wrap_process.o
165 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include slave/slave
        .c -o slave/slave.o
166 [exec] g++ -c -g -fPIC -DLINUX -Wall -I. -I../../include slave/
        wrap_process.c -o slave/wrap_process.o
167 [exec] g++ -pthread -shared -Wl,-soname,libslave.so -o libslave.so
        ../../util/*.o slave/*.o *.o
168 [exec] mv libslave.so "../../lib/Controller/libslave.so"
169 [exec] g++ -pthread -shared -Wl,-soname,libmaster.so -o libmaster.so
         ../../util/*.o master/*.o *.o
170 [exec] mv libmaster.so "../../lib/Controller/libmaster.so"
171 [exec] make[1]: Leaving directory '/home/user/dolc_ethz/build/bin/
        main/exampleDemo/dal-src/singlelinux/app/Controller'
```

```
172
173 [echo] Run singlelinux application.
174 [concat] periodic5 Cntr: register next activation at 0
175 [concat] periodic6 Cntr: register next activation at 0
176 [concat] periodic1 Cntr: register next activation at 0
177 [concat] periodic2 Cntr: register next activation at 0
178 [concat] periodic3 Cntr: register next activation at 0
179 [concat] periodic5 Cntr: fire at time 0
180 [concat] Periodic 5: Activated with index 0, no new data available
        from periodic1
181 [concat] Periodic 5: Activated with index 0, no new data available
        from periodic2
182 [concat] Periodic 5: Activated with index 0, no new data available
        from periodic1
183 [concat] Periodic 5: Activated with index 0, no new data available
        from periodic4
184 [concat] Periodic 5: Activated with index 0, sent data 0
185 [concat] periodic5 Cntr: register next activation at 200
186 [concat] periodic6 Cntr: fire at time 0
187 [concat] Periodic 6: Activated with index 0, read data: 0
188 [concat] periodic6 Cntr: register next activation at 200
189 [concat] periodic1 Cntr: fire at time 0
190 [concat] Periodic 1: Activated with index 0, sent data 0
191 [concat] periodic1 Cntr: register next activation at 100
192 [concat] periodic2 Cntr: fire at time 0
193 [concat] Periodic 2: Activated with index 0, sent data 0
194 [concat] periodic2 Cntr: register next activation at 100
195 [concat] periodic3 Cntr: fire at time 0
196 [concat] Periodic 3: Activated with index 0, sent data 0
197 [concat] periodic3 Cntr: register next activation at 100
198 [concat] periodic4 Cntr: register next activation at 0
199 [concat] periodic4 Cntr: fire at time 0
200 [concat] Periodic 4: Activated with index 0, sent data 0
201 [concat] periodic1 Cntr: fire at time 100
202 [concat] Periodic 1: Activated with index 1, sent data 1
203 [concat] periodic1 Cntr: register next activation at 200
204 [concat] periodic2 Cntr: fire at time 100
205 [concat] Periodic 2: Activated with index 1, sent data 1
206 [concat] periodic2 Cntr: register next activation at 200
207 [concat] periodic4 Cntr: register next activation at 100
208 [concat] periodic3 Cntr: fire at time 100
209 [concat] Periodic 3: Activated with index 1, sent data 1
210 [concat] periodic3 Cntr: register next activation at 200
211 [concat] periodic4 Cntr: fire at time 100
212 [concat] Periodic 4: Activated with index 1, sent data 1
213 [concat] periodic5 Cntr: fire at time 200
214 [concat] Periodic 5: Activated with index 1, read data from
        periodic1: 0
215 [concat] Periodic 5: Activated with index 1, read data from
        periodic2: 0
216 [concat] Periodic 5: Activated with index 1, read data from
        periodic3: 0
217 [concat] Periodic 5: Activated with index 1, read data from
        periodic4: 0
218 [concat] Periodic 5: Activated with index 1, sent data 4
219 [concat] periodic5 Cntr: register next activation at 400
220 [concat] periodic6 Cntr: fire at time 200
221 [concat] Periodic 6: Activated with index 1, read data: 4
222 [concat] periodic6 Cntr: register next activation at 400
223 [concat] periodic1 Cntr: fire at time 200
224 [concat] Periodic 1: Activated with index 2, sent data 2
225 [concat] periodic1 Cntr: register next activation at 300
```

```
226 [concat] periodic2 Cntr: fire at time 200
227 [concat] Periodic 2: Activated with index 2, sent data 2
228 [concat] periodic2 Cntr: register next activation at 300
229 [concat] periodic4 Cntr: register next activation at 200
230 [concat] periodic3 Cntr: fire at time 200
231 [concat] Periodic 3: Activated with index 2, sent data 2
232 [concat] periodic3 Cntr: register next activation at 300
233 [concat] periodic4 Cntr: fire at time 200
234 [concat] Periodic 4: Activated with index 2, sent data 2
235 [concat] periodic1 Cntr: fire at time 300
236 [concat] Periodic 1: Activated with index 3, sent data 3
237 [concat] periodic1 Cntr: register next activation at 400
238 [concat] periodic2 Cntr: fire at time 300
239 [concat] Periodic 2: Activated with index 3, sent data 3
240 [concat] periodic2 Cntr: register next activation at 400
241 [concat] periodic4 Cntr: register next activation at 300
242 [concat] periodic3 Cntr: fire at time 300
243 [concat] Periodic 3: Activated with index 3, sent data 3
244 [concat] periodic3 Cntr: register next activation at 400
245 [concat] periodic4 Cntr: fire at time 300
246 [concat] Periodic 4: Activated with index 3, sent data 3
247 [concat] periodic5 Cntr: fire at time 400
248 [concat] Periodic 5: Activated with index 2, read data from
        periodic1: 1
249 [concat] Periodic 5: Activated with index 2, read data from
        periodic2: 1
250 [concat] Periodic 5: Activated with index 2, read data from
        periodic3: 1
251 [concat] Periodic 5: Activated with index 2, read data from
        periodic4: 1
252 [concat] Periodic 5: Activated with index 2, sent data 4
253 [concat] periodic5 Cntr: register next activation at 600
254 [concat] periodic6 Cntr: fire at time 400
255 [concat] Periodic 6: Activated with index 2, read data: 4
256 [concat] periodic6 Cntr: register next activation at 600
257 [concat] periodic1 Cntr: fire at time 400
258 [concat] Periodic 1: Activated with index 4, sent data 4
259 [concat] periodic1 Cntr: register next activation at 500
260 [concat] periodic2 Cntr: fire at time 400
261 [concat] Periodic 2: Activated with index 4, sent data 4
262 [concat] periodic2 Cntr: register next activation at 500
263 [concat] periodic4 Cntr: register next activation at 400
264 [concat] periodic3 Cntr: fire at time 400
265 [concat] Periodic 3: Activated with index 4, sent data 4
266 [concat] periodic3 Cntr: register next activation at 500
267 [concat] periodic4 Cntr: fire at time 400
268 [concat] Periodic 4: Activated with index 4, sent data 4
269 [concat] periodic1 Cntr: fire at time 500
270 [concat] Periodic 1: Activated with index 5, sent data 5
271 [concat] periodic1 Cntr: register next activation at 600
272 [concat] periodic2 Cntr: fire at time 500
273 [concat] Periodic 2: Activated with index 5, sent data 5
274 [concat] periodic2 Cntr: register next activation at 600
275 [concat] periodic4 Cntr: register next activation at 500
276 [concat] periodic3 Cntr: fire at time 500
277 [concat] Periodic 3: Activated with index 5, sent data 5
278 [concat] periodic3 Cntr: register next activation at 600
279 [concat] periodic4 Cntr: fire at time 500
280 [concat] Periodic 4: Activated with index 5, sent data 5
281 [concat] periodic5 Cntr: fire at time 600
282 [concat] Periodic 5: Activated with index 3, read data from
        periodic1: 2
```

```
283 [concat] Periodic 5: Activated with index 3, read data from
        periodic2: 2
284 [concat] Periodic 5: Activated with index 3, read data from
        periodic3: 2
285 [concat] Periodic 5: Activated with index 3, read data from
        periodic4: 2
286 [concat] Periodic 5: Activated with index 3, sent data 4
287 [concat] periodic5 Cntr: register next activation at 800
288 [concat] periodic6 Cntr: fire at time 600
289 [concat] Periodic 6: Activated with index 3, read data: 4
290 [concat] periodic6 Cntr: register next activation at 800
291 [concat] periodic1 Cntr: fire at time 600
292 [concat] Periodic 1: Activated with index 6, sent data 6
293 [concat] periodic1 Cntr: register next activation at 700
294 [concat] periodic2 Cntr: fire at time 600
295 [concat] Periodic 2: Activated with index 6, sent data 6
296 [concat] periodic2 Cntr: register next activation at 700
297 [concat] periodic4 Cntr: register next activation at 600
298 [concat] periodic3 Cntr: fire at time 600
299 [concat] Periodic 3: Activated with index 6, sent data 6
300 [concat] periodic3 Cntr: register next activation at 700
301 [concat] periodic4 Cntr: fire at time 600
302 [concat] Periodic 4: Activated with index 6, sent data 6
303 [concat] periodic1 Cntr: fire at time 700
304 [concat] Periodic 1: Activated with index 7, sent data 7
305 [concat] periodic1 Cntr: register next activation at 800
306 [concat] periodic2 Cntr: fire at time 700
307 [concat] Periodic 2: Activated with index 7, sent data 7
308 [concat] periodic2 Cntr: register next activation at 800
309 [concat] periodic4 Cntr: register next activation at 700
310 [concat] periodic3 Cntr: fire at time 700
311 [concat] Periodic 3: Activated with index 7, sent data 7
312 [concat] periodic3 Cntr: register next activation at 800
313 [concat] periodic4 Cntr: fire at time 700
314 [concat] Periodic 4: Activated with index 7, sent data 7
315 [concat] periodic5 Cntr: fire at time 800
316 [concat] Periodic 5: Activated with index 4, read data from
        periodic1: 3
317 [concat] Periodic 5: Activated with index 4, read data from
        periodic2: 3
318 [concat] Periodic 5: Activated with index 4, read data from
        periodic3: 3
319 [concat] Periodic 5: Activated with index 4, read data from
        periodic4: 3
320 [concat] Periodic 5: Activated with index 4, sent data 4
321 [concat] periodic5 Cntr: register next activation at 1000
322 [concat] periodic6 Cntr: fire at time 800
323 [concat] Periodic 6: Activated with index 4, read data: 4
324 [concat] periodic6 Cntr: register next activation at 1000
325 [concat] periodic1 Cntr: fire at time 800
326 [concat] Periodic 1: Activated with index 8, sent data 8
327 [concat] periodic1 Cntr: register next activation at 900
328 [concat] periodic2 Cntr: fire at time 800
329 [concat] Periodic 2: Activated with index 8, sent data 8
330 [concat] periodic2 Cntr: register next activation at 900
331 [concat] periodic4 Cntr: register next activation at 800
332 [concat] periodic3 Cntr: fire at time 800
333 [concat] Periodic 3: Activated with index 8, sent data 8
334 [concat] periodic3 Cntr: register next activation at 900
335 [concat] periodic4 Cntr: fire at time 800
336 [concat] Periodic 4: Activated with index 8, sent data 8
337 [concat] periodic1 Cntr: fire at time 900
```

```
[concat] Periodic 1: Activated with index 9, sent data 9
[concat] periodic1 Cntr: register next activation at 1000
[concat] periodic2 Cntr: fire at time 900
[concat] Periodic 2: Activated with index 9, sent data 9
[concat] periodic2 Cntr: register next activation at 1000
[concat] periodic4 Cntr: register next activation at 900
[concat] periodic3 Cntr: fire at time 900
[concat] Periodic 3: Activated with index 9, sent data 9
[concat] periodic3 Cntr: register next activation at 1000
[concat] periodic4 Cntr: fire at time 900
[concat] Periodic 4: Activated with index 9, sent data 9
[concat] periodic5 Cntr: fire at time 1000
[concat] Periodic 5: Activated with index 5, read data from
    periodic1: 4
[concat] Periodic 5: Activated with index 5, read data from
    periodic2: 4
[concat] Periodic 5: Activated with index 5, read data from
    periodic3: 4
[concat] Periodic 5: Activated with index 5, read data from
    periodic4: 4
[concat] Periodic 5: Activated with index 5, sent data 4
[concat] periodic5 Cntr: register next activation at 1200
[concat] periodic6 Cntr: fire at time 1000
[concat] Periodic 6: Activated with index 5, read data: 4
[concat] periodic6 Cntr: register next activation at 1200
[concat] periodic1 Cntr: fire at time 1000
[concat] Periodic 1: Activated with index 10, sent data 10
[concat] periodic1 Cntr: register next activation at 1100
[concat] periodic2 Cntr: fire at time 1000
[concat] Periodic 2: Activated with index 10, sent data 10
[concat] periodic2 Cntr: register next activation at 1100
[concat] periodic4 Cntr: register next activation at 1000
[concat] periodic3 Cntr: fire at time 1000
[concat] Periodic 3: Activated with index 10, sent data 10
[concat] periodic3 Cntr: register next activation at 1100
[concat] periodic4 Cntr: fire at time 1000
[concat] Periodic 4: Activated with index 10, sent data 10
[concat] periodic1 Cntr: fire at time 1100
[concat] Periodic 1: Activated with index 11, sent data 11
[concat] periodic1 Cntr: register next activation at 1200
[concat] periodic2 Cntr: fire at time 1100
[concat] Periodic 2: Activated with index 11, sent data 11
[concat] periodic2 Cntr: register next activation at 1200
[concat] periodic4 Cntr: register next activation at 1100
[concat] periodic3 Cntr: fire at time 1100
[concat] Periodic 3: Activated with index 11, sent data 11
[concat] periodic3 Cntr: register next activation at 1200
[concat] periodic4 Cntr: fire at time 1100
[concat] Periodic 4: Activated with index 11, sent data 11
[concat] periodic5 Cntr: fire at time 1200
[concat] Periodic 5: Activated with index 6, read data from
    periodic1: 5
[concat] Periodic 5: Activated with index 6, read data from
    periodic2: 5
[concat] Periodic 5: Activated with index 6, read data from
    periodic3: 5
[concat] Periodic 5: Activated with index 6, read data from
    periodic4: 5
[concat] Periodic 5: Activated with index 6, sent data 4
[concat] periodic5 Cntr: register next activation at 1400
[concat] periodic6 Cntr: fire at time 1200
[concat] Periodic 6: Activated with index 6, read data: 4
```

```
[concat] periodic6 Cntr: register next activation at 1400
[concat] periodic1 Cntr: fire at time 1200
[concat] Periodic 1: Activated with index 12, sent data 12
[concat] periodic1 Cntr: register next activation at 1300
[concat] periodic2 Cntr: fire at time 1200
[concat] Periodic 2: Activated with index 12, sent data 12
[concat] periodic2 Cntr: register next activation at 1300
[concat] periodic4 Cntr: register next activation at 1200
[concat] periodic3 Cntr: fire at time 1200
[concat] Periodic 3: Activated with index 12, sent data 12
[concat] periodic3 Cntr: register next activation at 1300
[concat] periodic4 Cntr: fire at time 1200
[concat] Periodic 4: Activated with index 12, sent data 12
[concat] periodic1 Cntr: fire at time 1300
[concat] Periodic 1: Activated with index 13, sent data 13
[concat] periodic1 Cntr: register next activation at 1400
[concat] periodic2 Cntr: fire at time 1300
[concat] Periodic 2: Activated with index 13, sent data 13
[concat] periodic2 Cntr: register next activation at 1400
[concat] periodic4 Cntr: register next activation at 1300
[concat] periodic3 Cntr: fire at time 1300
[concat] Periodic 3: Activated with index 13, sent data 13
[concat] periodic3 Cntr: register next activation at 1400
[concat] periodic4 Cntr: fire at time 1300
[concat] Periodic 4: Activated with index 13, sent data 13
[concat] periodic5 Cntr: fire at time 1400
[concat] Periodic 5: Activated with index 7, read data from
    periodic1: 6
[concat] Periodic 5: Activated with index 7, read data from
    periodic2: 6
[concat] Periodic 5: Activated with index 7, read data from
    periodic3: 6
[concat] Periodic 5: Activated with index 7, read data from
    periodic4: 6
[concat] Periodic 5: Activated with index 7, sent data 4
[concat] periodic5 Cntr: register next activation at 1600
[concat] periodic6 Cntr: fire at time 1400
[concat] Periodic 6: Activated with index 7, read data: 4
[concat] periodic6 Cntr: register next activation at 1600
[concat] periodic1 Cntr: fire at time 1400
[concat] Periodic 1: Activated with index 14, sent data 14
[concat] periodic1 Cntr: register next activation at 1500
[concat] periodic2 Cntr: fire at time 1400
[concat] Periodic 2: Activated with index 14, sent data 14
[concat] periodic2 Cntr: register next activation at 1500
[concat] periodic4 Cntr: register next activation at 1400
[concat] periodic3 Cntr: fire at time 1400
[concat] Periodic 3: Activated with index 14, sent data 14
[concat] periodic3 Cntr: register next activation at 1500
[concat] periodic4 Cntr: fire at time 1400
[concat] Periodic 4: Activated with index 14, sent data 14
[concat] periodic1 Cntr: fire at time 1500
[concat] Periodic 1: Activated with index 15, sent data 15
[concat] periodic1 Cntr: register next activation at 1600
[concat] periodic2 Cntr: fire at time 1500
[concat] Periodic 2: Activated with index 15, sent data 15
[concat] periodic2 Cntr: register next activation at 1600
[concat] periodic4 Cntr: register next activation at 1500
[concat] periodic3 Cntr: fire at time 1500
[concat] Periodic 3: Activated with index 15, sent data 15
[concat] periodic3 Cntr: register next activation at 1600
[concat] periodic4 Cntr: fire at time 1500
```

```
[concat] Periodic 4: Activated with index 15, sent data 15
[concat] periodic5 Cntr: fire at time 1600
[concat] Periodic 5: Activated with index 8, read data from
    periodic1: 7
[concat] Periodic 5: Activated with index 8, read data from
    periodic2: 7
[concat] Periodic 5: Activated with index 8, read data from
    periodic3: 7
[concat] Periodic 5: Activated with index 8, read data from
    periodic4: 7
[concat] Periodic 5: Activated with index 8, sent data 4
[concat] periodic5 Cntr: register next activation at 1800
[concat] periodic6 Cntr: fire at time 1600
[concat] Periodic 6: Activated with index 8, read data: 4
[concat] periodic6 Cntr: register next activation at 1800
[concat] periodic1 Cntr: fire at time 1600
[concat] Periodic 1: Activated with index 16, sent data 16
[concat] periodic1 Cntr: register next activation at 1700
[concat] periodic2 Cntr: fire at time 1600
[concat] Periodic 2: Activated with index 16, sent data 16
[concat] periodic2 Cntr: register next activation at 1700
[concat] periodic4 Cntr: register next activation at 1600
[concat] periodic3 Cntr: fire at time 1600
[concat] Periodic 3: Activated with index 16, sent data 16
[concat] periodic3 Cntr: register next activation at 1700
[concat] periodic4 Cntr: fire at time 1600
[concat] Periodic 4: Activated with index 16, sent data 16
[concat] periodic1 Cntr: fire at time 1700
[concat] Periodic 1: Activated with index 17, sent data 17
[concat] periodic1 Cntr: register next activation at 1800
[concat] periodic2 Cntr: fire at time 1700
[concat] Periodic 2: Activated with index 17, sent data 17
[concat] periodic2 Cntr: register next activation at 1800
[concat] periodic4 Cntr: register next activation at 1700
[concat] periodic3 Cntr: fire at time 1700
[concat] Periodic 3: Activated with index 17, sent data 17
[concat] periodic3 Cntr: register next activation at 1800
[concat] periodic4 Cntr: fire at time 1700
[concat] Periodic 4: Activated with index 17, sent data 17
[concat] periodic5 Cntr: fire at time 1800
[concat] Periodic 5: Activated with index 9, read data from
    periodic1: 8
[concat] Periodic 5: Activated with index 9, read data from
    periodic2: 8
[concat] Periodic 5: Activated with index 9, read data from
    periodic3: 8
[concat] Periodic 5: Activated with index 9, read data from
    periodic4: 8
[concat] Periodic 5: Activated with index 9, sent data 4
[concat] periodic5 Cntr: register next activation at 2000
[concat] periodic6 Cntr: fire at time 1800
[concat] Periodic 6: Activated with index 9, read data: 4
[concat] periodic6 Cntr: register next activation at 2000
[concat] periodic1 Cntr: fire at time 1800
[concat] Periodic 1: Activated with index 18, sent data 18
[concat] periodic1 Cntr: register next activation at 1900
[concat] periodic2 Cntr: fire at time 1800
[concat] Periodic 2: Activated with index 18, sent data 18
[concat] periodic2 Cntr: register next activation at 1900
[concat] periodic4 Cntr: register next activation at 1800
[concat] periodic3 Cntr: fire at time 1800
[concat] Periodic 3: Activated with index 18, sent data 18
```

```
504 [concat] periodic3 Cntr: register next activation at 1900
505 [concat] periodic4 Cntr: fire at time 1800
506 [concat] Periodic 4: Activated with index 18, sent data 18
507 [concat] periodic1 Cntr: fire at time 1900
508 [concat] Periodic 1: Activated with index 19, sent data 19
509 [concat] periodic1 Cntr: register next activation at 2000
510 [concat] periodic2 Cntr: fire at time 1900
511 [concat] Periodic 2: Activated with index 19, sent data 19
512 [concat] periodic2 Cntr: register next activation at 2000
513 [concat] periodic4 Cntr: register next activation at 1900
514 [concat] periodic3 Cntr: fire at time 1900
515 [concat] Periodic 3: Activated with index 19, sent data 19
516 [concat] periodic3 Cntr: register next activation at 2000
517 [concat] periodic4 Cntr: fire at time 1900
518 [concat] Periodic 4: Activated with index 19, sent data 19
519 [concat] periodic5 Cntr: fire at time 2000
520 [concat] Periodic 5: Activated with index 10, read data from
        periodic1: 9
521 [concat] Periodic 5: Activated with index 10, read data from
        periodic2: 9
522 [concat] Periodic 5: Activated with index 10, read data from
        periodic3: 9
523 [concat] Periodic 5: Activated with index 10, read data from
        periodic4: 9
524 [concat] Periodic 5: Activated with index 10, sent data 4
525 [concat] periodic5 Cntr: register next activation at 2200
526 [concat] periodic6 Cntr: fire at time 2000
527 [concat] Periodic 6: Activated with index 10, read data: 4
528 [concat] periodic6 Cntr: register next activation at 2200
529 [concat] periodic1 Cntr: fire at time 2000
530 [concat] Periodic 1: Activated with index 20, sent data 20
531 [concat] periodic1 Cntr: register next activation at 2100
532 [concat] periodic2 Cntr: fire at time 2000
533 [concat] Periodic 2: Activated with index 20, sent data 20
534 [concat] periodic2 Cntr: register next activation at 2100
535 [concat] periodic4 Cntr: register next activation at 2000
536 [concat] periodic3 Cntr: fire at time 2000
537 [concat] Periodic 3: Activated with index 20, sent data 20
538 [concat] periodic3 Cntr: register next activation at 2100
539 [concat] periodic4 Cntr: fire at time 2000
540 [concat] Periodic 4: Activated with index 20, sent data 20
541 [concat] periodic4 Cntr: register next activation at 2100
542
543 mapping-optimization:
544 [java] Read architecture from XML file
545 [java] -- full filename: file:/home/user/dolc_ethz/examples/arch/
        archDemo.xml
546 [java] -- Architecture model from XML [Finished]
547 [java]
548 [java] Read application from XML file
549 [java] -- full filename: file: exampleDemo/exampleDemo-DOLC.xml
550 [java] -- Application model from XML [Finished]
551 [java]
552 [java] Start mapping and scheduling optimization
553 [java] -- Initial objective: 0.043493596216835996
554 [java] -- Best objective: 0.04020293282578188
555 [java] -- Examined variations: 17161
556 [java] -- Admissible: true
557 [java] -- Required time: 438 msec
558 [java]
559 [java] Generate mapping XML
```

```
560 [java] -- Optimized mapping has been saved in exampleDemo/mapping.
        xml
561
562 BUILD SUCCESSFUL
563 Total time: 14 seconds
```

6 Accurate Timing Analysis under Resource Contention

For specific TTS mapping solutions (e.g., for the outcome of the mapping optimization procedure), it is possible to perform timing analysis using a timed automata representation of the system in order to obtain accurate estimations of the worst-case TTS sub-frame lengths. This enables, in turn, the accurate estimation of the spare time which remains at the end of each TTS frame, which can be later used if more tasks need to be scheduled in the system. The methodology for the timing analysis under resource contention scenarios is detailed in the deliverable D5.1 Interference Analysis and Isolation Mechanisms. In order to apply it, the following tools are required:

- **Uppaal:** http://uppaal.org/ (version 4.1.7 or greater)
- **Matlab:** http://www.mathworks.ch/products/matlab/ (version R2011a or greater)

An example on how to apply timing / interference analysis, using the timed automata models under `dolc_ethz/analysis/`, is given in the DOL-Critical web page: http://www.tik.ee.ethz.ch/∼certainty/dolc.html.

7 Future Extensions & Troubleshooting

In the current version of DOL-Critical, the mapping optimizer may behave unexpectedly in the following cases. These cases are either not supported or they have not been tested thoroughly.

- Applications with more than two criticality levels: So far only applications with two criticality levels (DAL-B, DAL-C or DAL-C, DAL-D) have been tested.
- Applications with criticality levels other than DAL-B, DAL-C, DAL-D: An error message will be printed (outside FMS scope).
- Precedence constraints among tasks with non-equal periods: An error message will be printed and the period of the first task of the precedence chain will be taken as the common period.
- Precedence constraints from lower-criticality tasks to higher criticality tasks: An error message will be printed and the precedence constraint may not hold if both tasks are scheduled in the same TTS frame in the optimized mapping solution.

- Application tasks with more than one `superblock` element for the same execution mode: It has not been tested thoroughly.
- Application tasks (restartable) with values other than 1 for parameters `minRep, maxRep`: Not supported yet.
- Application task controllers with `period` values smaller than 1 us: It may lead to numerical errors upon dimensioning of the TTS cycle.
- Architecture specifications with more than one `shared` (memory) elements: An error message will be printed and only the first `shared` element will be considered.
- Architecture specifications with arbitration policy other than `roundrobin` for the `shared` element: An error message will be printed and round-robin arbitration will be considered.
- Architecture specifications with complex bus hierarchies and / or `noc` elements: Not supported (outside MPPA-256 single-cluster scope).
- TTS dimensioning with non-equally sized frames: Not supported yet.

If you encounter any problem upon experimenting with new example applications and / or architectures, please do not hesitate to contact us (certainty@tik.ee.ethz.ch).

8 Archive Structure

The DOL-Critical archive includes the following folders:

- **documentation/**
  - **DOL-Critical technical reports/**: Tool chain overview, XML semantics, C/C++ coding style (programming interface).
  - **publications/**: All related publications concerning the Time-Triggered scheduling with Synchronization points (TTS), mapping optimization, and timing analysis under resource contention scenarios.
- **dolc_ethz/**
  - **analysis/**: Sample timed automata models of the system for timing analysis under resource contention.
  - **bin/**: DOL-Critical library (dolc.jar) and required third-party libraries.
  - **examples/**: Sample application (XML, C code) and architecture (XML) specifications.

    - **arch/**: 2-core (`archDemo.xml`) and 4-core (`inteli7620M_onecluster.xml`) sample architectures.
    - **exampleDemo/**: 6-task application, detailed in http://www.tik.ee.ethz.ch/∼certainty/dolc.html.
    - **exampleProducerConsumer/**: 2-task producer-consumer application. The producer writes every 200 ms to a mailbox, the contents of which are read out by the consumer every 100 ms.
    - **exampleModeSwitch/**: 2-task producer-consumer application. Similar as above with equal read/write rate. The producer changes the execution mode of the consumer at every fifth activation.
    - **exampleArrayErase/**: Similar as exampleModeSwitch. The producer and consumer write / read two elements at each activation. At every fifth activation, the producer not only changes the execution mode of the consumer, but it also erases the last two written elements.
    - **runexample.xml**: Ant files for automated execution of examples.
- **schema/**: XML semantics definition, used for XML parsing and consistency checks.
- **build_zip.xml**: ANT file required for installation of DOL-Critical.
- **README**: Short guide to DOL-Critical installation and demo example execution.