Periodic Task Mining in Embedded System Traces

Oleg Iegorov  Reinier Torres  Sebastian Fischmeister

Real-Time Embedded System Group

April 20, 2017
PeTaMi (PEriodic TAsk MIner)

A solution for reverse engineering Real-Time Embedded Systems (RTES):
• standalone tool, or
• part of a comprehensive framework

Real-Time Embedded System

execution

Trace

Periodic tasks
• task_5D3: T=4ms, RTP=[1.3ms, 0.8ms]
• task_37A: T=12ms, RTP=[5ms]
• task_1Z1: T=8ms, RTP=[2ms, 0.4ms, 1.5ms]
...
PeTaMi (PEriodic TAsk MIner)

A solution for **reverse engineering** Real-Time Embedded Systems (RTES):

- standalone tool, or
- part of a comprehensive framework
"(Software) reverse engineering is the process of analyzing a subject system to create representations of the system at a higher level of abstraction" \(^1\)

- Traditionally used in the domain of desktop and enterprise software

Software Growth in Real-time Systems

Lines of Code

Year

Sources:

- Charette RN. "This car runs on code". IEEE spectrum. 2009 Feb 1;46(3):3.
Complex RTES

Need of software models for:

- legacy code maintenance
- debugging
- testing.
Need of software models for:

- legacy code maintenance
- debugging
- testing.

Problems:

- Software models usually not available. (tight time-to-market, constant changes to code)
- State-of-the-art tools do not deal with timeliness.
Complex RTES

Need of software models for:
- legacy code maintenance
- debugging
- testing.

Problems:
- Software models usually not available. 
  (tight time-to-market, constant changes to code)
- State-of-the-art tools do not deal with timeliness.

Reverse engineering tools tailored for RTES are needed
Characteristics of a Reverse Engineering Tool

- Representation of a system
  - Finite state machine;
  - Petri net;
  - Regular expressions;
  - UML model;
  - and others...

- Extraction of information from a system
  - Static: from source code;
  - Dynamic: from traces;
  - Hybrid.
Characteristics of a Reverse Engineering Tool

- Representation of a system
  - Finite state machine;
  - Petri net;
  - Regular expressions;
  - UML model;
  - Task set; ← PeTaMi
  - and others...

- Extraction of information from a system
  - Static: from source code;
  - Dynamic: from traces;
  - Hybrid.
Characteristics of a Reverse Engineering Tool

- Representation of a system
  - Finite state machine;
  - Petri net;
  - Regular expressions;
  - UML model;
  - Task set; ← PeTaMi
  - and others...

- Extraction of information from a system
  - Static: from source code;
  - Dynamic: from traces; ← PeTaMi
  - Hybrid.
PeTaMi (PEriodic TAsk MIner)

Real-Time Embedded System

Trace

Periodic tasks

- task_5D3: $T=4$ ms, RTP=[1.3 ms, 0.8 ms]
- task_37A: $T=12$ ms, RTP=[5 ms]
- task_1Z1: $T=8$ ms, RTP=[2 ms, 0.4 ms, 1.5 ms]
...
RTES Execution Trace

timestamp  fields  ...  event

```
248895958208,00,THREAD,THRUNNING,9,1,................,24,2,,",",proc/b
248889591375,00,THREAD,THREADY,1,1,................,0,1,,",",proc/boot
24889596458,00,COMM,REC_PULSE,9,................,0x40000003,................,"",pr
24889614208,00,THREAD,THRREIVE,9,1,................,24,2,"",proc/b
24889615751,00,THREAD,THRUNNING,1,1,................,0,1,"",proc/boot
24889982541,00,COMM,SND_PULSE_EXE,8,................,0x40000003,................,"",pr
24889983833,00,THREAD,THRREIVE,8,1,................,24,2,"",proc/b
24889985041,00,THREAD,THREADY,1,1,................,0,1,"",proc/boot
24889989708,00,COMM,REC_PULSE,8,................,0x40000003,................,"",pr
24890001916,00,THREAD,THRREIVE,8,1,................,24,2,"",proc/b
24890003458,00,THREAD,THRUNNING,1,1,................,0,1,"",proc/bo
24890368583,00,COMM,SND_PULSE_EXE,61465,................,0x40000006,
24890370000,00,COMM,SND_PULSE_EXE,61465,................,0x4000000c
24890371291,00,THREAD,THRREIVE,61465,4,................,10,2,"",tm
24890372541,00,THREAD,THREADY,1,1,................,0,1,"",proc/boot
24890373916,00,COMM,SND_PULSE_EXE,61465,................,0x40000009,
24890377916,00,COMM,REC_PULSE,61465,................,0x4000000c
24890431666,00,THREAD,THRREIVE,61465,4,................,10,2,"",tm
24890433208,00,THREAD,THRUNNING,1,1,................,0,1,"",proc/boot
```
RTES Execution Trace

- **Trace**: chronologically ordered list of trace entries;
- **Trace entry**: tuple \(<\text{timestamp}, \text{task ID}\>\);
- **Task ID**: conjunction of fields that uniquely identify the task.
A Real-Time Task

A timeline view of execution of a real-time task on a CPU:

- blue boxes: CPU is executing the task;
- yellow boxes: CPU is idle or executing other tasks.
A Real-Time Task

- **Task**: a recurrent program that executes a sequence of jobs;
- **Periodic task** releases jobs with a fixed interval of time;
- **Response time**: time between when a task’s job is released and when it is completed;

![Diagram of real-time task with response time and period (P) markings.](image)
A Real-Time Task

Trace entries of a real-time task:

- nonpreemptive OS: one per job;
A Real-Time Task

Trace entries of a real-time task:

- nonpreemptive OS: one per job;
- preemptive OS: one or more per job;
Problem: detect if a task is periodic from its trace entries.
- **Inter-Arrival Time (IAT)**: between consecutive task invocations;

- **Intra-Job Interval (IJI)**: between consecutive task executions within a job;

- **Between-Job Interval (BJI)**: between when the task finished executing its job and the first task invocation from the next job;

- **Whole-Job Interval (WJI)**: between first task invocations of two consecutive jobs.

WJIs of a periodic task have the same value. WJIs can be extracted from BJIs.

**Problem:** extract BJIs from IATs.
Terminology

- Inter-Arrival Time (IAT): between consecutive task invocations;
- Intra-Job Interval (IJI): between consecutive task executions within a job;
- Between-Job Interval (BJI): between when the task finished executing its job and the first task invocation from the next job;
- Whole-Job Interval (WJI): between first task invocations of two consecutive jobs.

WJIs of a periodic task have \( \sim \) same value. WJIs can be extracted from BJIs.

Problem: extract BJIs from IATs.
Terminology

- **Inter-Arrival Time (IAT)**: between consecutive task invocations;
- **Intra-Job Interval (IJI)**: between consecutive task executions within a job;
- **Between-Job Interval (BJI)**: between when the task finished executing its job and the first task invocation from the next job;
- **Whole-Job Interval (WJI)**: between first task invocations of two consecutive jobs.

WJIs of a periodic task have the same value. WJIs can be extracted from BJIs.

Problem: extract BJIs from IATs.
Terminology

- **Inter-Arrival Time (IAT)**: between consecutive task invocations;
- **Intra-Job Interval (IJI)**: between consecutive task executions within a job;
- **Between-Job Interval (BJI)**: between when the task finished executing its job and the first task invocation from the next job;
- **Whole-Job Interval (WJI)**: between first task invocations of two consecutive jobs.

BJIs can be extracted from IJIs.

**Problem:** extract BJIs from IATs.
Terminology

- **Inter-Arrival Time (IAT)**: between consecutive task invocations;
- **Intra-Job Interval (IJI)**: between consecutive task executions within a job;
- **Between-Job Interval (BJI)**: between when the task finished executing its job and the first task invocation from the next job;
- **Whole-Job Interval (WJI)**: between first task invocations of two consecutive jobs.

WJIs of a periodic task have the same value. WJIs can be extracted from BJIs.

Problem: extract BJIs from IATs.
Terminology

- **Inter-Arrival Time (IAT)**: between consecutive task invocations;
- **Intra-Job Interval (IJI)**: between consecutive task executions within a job;
- **Between-Job Interval (BJI)**: between when the task finished executing its job and the first task invocation from the next job;
- **Whole-Job Interval (WJI)**: between first task invocations of two consecutive jobs.

WJIs of a periodic task have $\sim$ same value.
Terminology

- **Inter-Arrival Time (IAT):** between consecutive task invocations;
- **Intra-Job Interval (IJI):** between consecutive task executions within a job;
- **Between-Job Interval (BJI):** between when the task finished executing its job and the first task invocation from the next job;
- **Whole-Job Interval (WJI):** between first task invocations of two consecutive jobs.

WJIs of a periodic task have $\sim$ same value.

WJIs can be extracted from BJIs.
Terminology

- **Inter-Arrival Time (IAT)**: between consecutive task invocations;
- **Intra-Job Interval (IJI)**: between consecutive task executions within a job;
- **Between-Job Interval (BJI)**: between when the task finished executing its job and the first task invocation from the next job;
- **Whole-Job Interval (WJI)**: between first task invocations of two consecutive jobs.

WJIs of a periodic task have \( \sim \) same value.

WJIs can be extracted from BJIs.

**Problem**: extract BJIs from IATs.
If we let a periodic task $\tau$ execute for a considerable amount of time, then the mean duration of $\tau$’s $BJI$s will be larger than the mean duration of $\tau$’s $IJI$s.
The Assumption

If we let a periodic task \( \tau \) execute for a considerable amount of time, then the mean duration of \( \tau \)'s BJIs will be larger than the mean duration of \( \tau \)'s IJIs.

Simulation:
- 1000 task sets;
- \( 10^7 \) time units;
- random \# of tasks per task set;
- random task’s execution time;
- utilization: \([0.5, 0.55, 0.6, \ldots, 1.0]\);
- periods computed with UUniFast algorithm \(^2\)

Unsafe RTES: CPU utilization > 70%.

---

Mining Periodic Tasks from RTES Traces

IAT

time

BJIs

IJIs
Mining Periodic Tasks from RTES Traces
Mining Periodic Tasks from RTES Traces

WJI=10

WJI=30

time

spread of WJIs

number of largest intervals, N

1  2  3  4  5  6  7  8
Mining Periodic Tasks from RTES Traces

WJI=10

WJI=20

WJI=10

8  7.5  6.5  5  4.5  2  1.5  1  1  0.5

spread of WJIs

1  2  3  4  5  6  7  8

number of largest intervals, N
Mining Periodic Tasks from RTES Traces

The diagram illustrates the mining of periodic tasks from RTES traces. The timeline is marked with intervals where the WJI (Workload Interval) is constant at 10. The tasks are represented by filled rectangles and the numbers inside indicate the task execution times. The spread of WJIs is shown in a scatter plot, with the number of largest intervals (N) on the x-axis and the spread of WJIs on the y-axis.

Time intervals: WJI=10

- Task execution times: 6.5, 8, 4.5, 5, 7.5

Spread of WJIs:

- Number of largest intervals, N:
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8

Graph showing the spread of WJIs and the number of largest intervals.
Mining Periodic Tasks from RTES Traces

![Diagram showing periodic tasks and their WJI intervals.

- **WJI=10** intervals:
  - First interval: 6.5 units
  - Second interval: 8 units

- **WJI=4** intervals:
  - Interval: 4.5 units
  - Interval: 2 units

- **WJI=6** interval:
  - Interval: 5 units

- **WJI=10** interval:
  - Interval: 7.5 units

The spread of WJIs is plotted against the number of largest intervals, N.

- Number of largest intervals:
  - 1: 8 units
  - 2: 7.5 units
  - 4: 6.5 units
  - 5: 5 units
  - 6: 4.5 units
  - 7: 2 units
  - 8: 1.5 units
  - 9: 1 unit
  - 10: 1 unit
  - 11: 0.5 units

The spread of WJIs is as follows:

- 1 interval: 8 units
- 2 intervals: 7.5 units
- 4 intervals: 6.5 units
- 5 intervals: 5 units
- 6 intervals: 4.5 units
- 7 intervals: 2 units
- 8 intervals: 1.5 units
- 9 intervals: 1 unit
- 10 intervals: 1 unit
- 11 intervals: 0.5 units
Mining Periodic Tasks from RTES Traces

Time intervals:
- 6.5
- 8
- 1.5
- 4.5
- 2
- 5
- 7.5

Number of largest intervals (N):
- 1
- 1
- 1
- 0.5

Spread of WJIs:
- 8
- 7.5
- 6.5
- 5
- 4.5

Time line with individual tasks and their durations.
Mining Periodic Tasks from RTES Traces

---

[Diagram showing time intervals and spread of WJIs]
Mining Periodic Tasks from RTES Traces

N largest intervals with the smallest spread of corresponding WJIs $\approx \{BJIs\}$
if \( \text{spread}(\text{WJIs}) < \text{threshold} \), then task is periodic;
period \( P = \text{mode}(\{\text{WJIs}\}) \)
Mining Response Time Profiles

\{\text{response times} \} = P - \{BJIs\}
Mining Response Time Profiles

\{\text{response times}\} = P - \{BJIs\}

Mining of the most common response times.
Mining Response Time Profiles

\[
\{\text{response times}\} = P - \{BJIs\}
\]

Mining of the most common response times.

Clustering problem: the gap statistic and k-medoids clustering.
1. Unmanned Aerial Vehicle:
   - QNX Neutrino 6.4 RTOS;
   - 8 execution traces: \( \sim 150K \) trace entries, 10 tasks;
   - accuracy of periodic task mining: 100%;
   - accuracy of RTP mining: 83%;
Case Studies

1. Unmanned Aerial Vehicle:
   - QNX Neutrino 6.4 RTOS;
   - 8 execution traces: \(\sim150K\) trace entries, 10 tasks;
   - accuracy of periodic task mining: 100%;
   - accuracy of RTP mining: 83%;

2. Toyota RAV4:
   - traces of Controller Area Network (CAN) messages;
   - 11 driving scenarios (starting the engine, lane changes, etc.);
   - \(\sim10K\) trace entries, 10-15 tasks per trace;
   - accuracy of periodic task mining: 100%;
Conclusion

PeTaMi reverse engineering tool:
- mines periodic tasks and their response time profiles from RTES execution traces;
- relies on the assumption that in safe systems the slack between task’s jobs is on average larger than job’s individual preemption times;
- achieves good accuracy on traces of industrial RTES.

![Real-Time Embedded System](image1)

![Trace](image2)

**Periodic tasks**
- task_5D3: T=4ms, RTP=[1.3ms, 0.8ms]
- task_37A: T=12ms, RTP=[5ms]
- task_1Z1: T=8ms, RTP=[2ms, 0.4ms, 1.5ms]
...
Slack time analysis