



# Exago

*Property monitoring via log file analysis*

Atilla Erdődi  
Eötvös Loránd University

Supervisor:  
Prof. John Derrick  
University of Sheffield

# About Exago

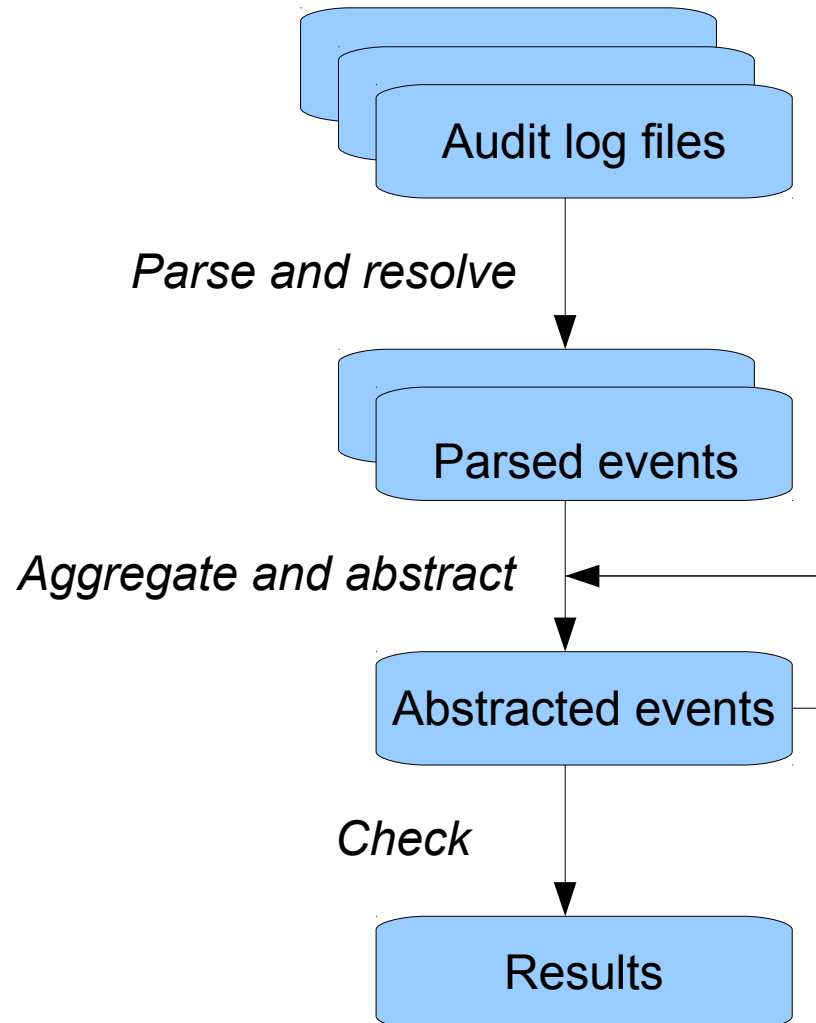
- *Post-mortem* monitoring tool written in Erlang
- Prototype by Hans Svensson,  
Chalmers University of Technology, Göteborg
- Developed by Erlang Solutions Ltd.



# Early results

- Two industrial case studies:
  - Found bugs in a system that has been in use for 3 years – *creating the model of the system took ~2 days!*
  - Has been used from the early stages of development – *providing quality assurance*

# The Approach



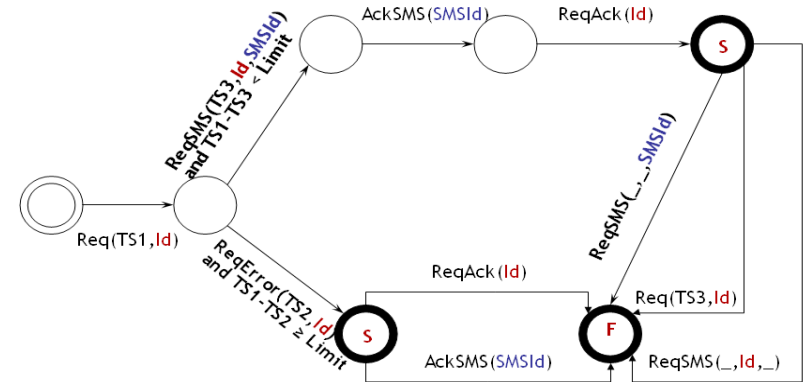
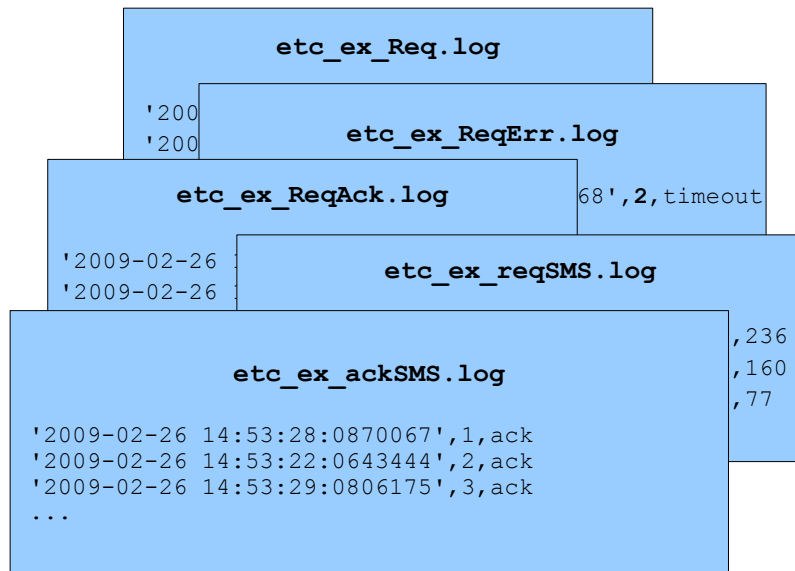
# Configuration

- Configuration as an Erlang proplist
- Specify the log files by wildcards
- Identify the relevant information from each entry (timestamp, session identifier, event details)
- Define the abstraction function(s)
- Define the abstract model of the system as a Finite State Machine



# Example

- A simplified, ideal SMS Gateway System
- CSV log files are artificially generated



# Parse and resolve

## etc\_ex\_Req.log

```
'2009-02-26 14:53:20:0000000',1  
'2009-02-26 14:53:20:0268786',2  
'2009-02-26 14:53:20:0531614',3  
'2009-02-26 14:53:21:0009257',4  
...
```

## etc\_ex\_ReqAck.log

```
'2009-02-26 14:53:29:0204458',1,ack  
'2009-02-26 14:53:22:0751754',3,ack  
'2009-02-26 14:53:29:0823714',4,ack  
...
```

## etc\_ex\_ReqErr.log

```
'2009-02-26 14:57:46:0474168',2,timeout  
...
```

## etc\_ex\_reqSMS.log

```
'2009-02-26 14:53:28:0701625',1,1,236  
'2009-02-26 14:53:22:0520858',3,2,160  
'2009-02-26 14:53:29:0529183',4,3,77  
...
```

## etc\_ex\_ackSMS.log

```
'2009-02-26 14:53:28:0870067',1,ack  
'2009-02-26 14:53:22:0643444',2,ack  
'2009-02-26 14:53:29:0806175',3,ack  
...
```

# Parse and resolve

## etc\_ex\_Req.log

```
'2009-02-26 14:53:20:0000000',1  
'2009-02-26 14:53:20:0268786',2  
'2009-02-26 14:53:20:0531614',3  
'2009-02-26 14:53:21:0009257',4  
...
```

## etc\_ex\_ReqAck.log

```
'2009-02-26 14:53:29:0204458',1,ack  
'2009-02-26 14:53:22:0751754',3,ack  
'2009-02-26 14:53:29:0823714',4,ack  
...
```

## etc\_ex\_ReqErr.log

```
'2009-02-26 14:57:46:0474168',2,timeout  
...
```

## etc\_ex\_reqSMS.log

```
'2009-02-26 14:53:28:0701625',1,1,236  
'2009-02-26 14:53:22:0520858',3,2,160  
'2009-02-26 14:53:29:0529183',4,3,77  
...
```

## etc\_ex\_ackSMS.log

```
'2009-02-26 14:53:28:0870067',1,ack  
'2009-02-26 14:53:22:0643444',2,ack  
'2009-02-26 14:53:29:0806175',3,ack  
...
```

# Parse and resolve

## etc\_ex\_Req.log

```
'2009-02-26 14:53:20:0000000',1  
'2009-02-26 14:53:20:0268786',2  
'2009-02-26 14:53:20:0531614',3  
'2009-02-26 14:53:21:0009257',4  
...
```

## etc\_ex\_ReqAck.log

```
'2009-02-26 14:53:29:0204458',1,ack
```

```
[{wildcard, "etc_ex_ReqErr*.log"},  
{parse_fun, csv},  
{parse_opts, [{delimiter, ","}]},  
{timestamp, [1]},  
{session_id, [2]},  
{abstract_value, [filename]}]
```

## etc\_ex\_ReqErr.log

```
'2009-02-26 14:57:46:0474168',2,timeout  
...
```

## etc\_ex\_ackSMS.log

```
'2009-02-26 14:53:28:0870067',1,ack  
'2009-02-26 14:53:22:0643444',2,ack  
'2009-02-26 14:53:29:0806175',3,ack  
...
```

# Parse and resolve

## etc\_ex\_Req.log

```
'2009-02-26 14:53:20:0000000',1  
'2009-02-26 14:53:20:0268786',2  
'2009-02-26 14:53:20:0531614',3  
'2009-02-26 14:53:21:0009257',4  
...
```

## etc\_ex\_ReqAck.log

```
'2009-02-26 14:53:29:0204458',1,ack  
'2009-02-26 14:53:22:0751754',3,ack  
'2009-02-26 14:53:29:0823714',4,ack  
...
```

## etc\_ex\_ReqErr.log

```
'2009-02-26 14:57:46:0474168',2,timeout  
...
```

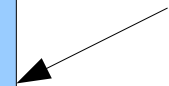
## etc\_ex\_reqSMS.log

```
'2009-02-26 14:53:28:0701625',1,1,236  
'2009-02-26 14:53:22:0520858',3,2,160  
'2009-02-26 14:53:29:0529183',4,3,77  
...
```

## etc\_ex\_ackSMS.log

```
'2009-02-26 14:53:28:0870067',1,ack  
'2009-02-26 14:53:22:0643444',2,ack  
'2009-02-26 14:53:29:0806175',3,ack  
...
```

No session id!



# Parse and resolve

```
etc_ex_reqSMS.log
'2009-02-26 14:53:28:0701625', 1, 1, 236
'2009-02-26 14:53:22:0520858', 3, 2, 160
'2009-02-26 14:53:29:0529183', 4, 3, 77
...
```

Defines a mapping between the ids!

```
[...,
 {mapping, [{sid, 3, 2}]},
 ...]
```

```
etc_ex_ackSMS.log
'2009-02-26 14:53:28:0870067', 1, ack
'2009-02-26 14:53:22:0643444', 2, ack
'2009-02-26 14:53:29:0806175', 3, ack
...
```

# Parse and resolve

```
etc_ex_reqSMS.log
'2009-02-26 14:53:28:0701625', 1, 1, 236
'2009-02-26 14:53:22:0520858', 3, 2, 160
'2009-02-26 14:53:29:0529183', 4, 3, 77
...
```

Defines a mapping between the ids!

```
[...,
 {mapping, [{sid, 3, 2}]},
 ...]
```

```
etc_ex_ackSMS.log
'2009-02-26 14:53:28:0870067', 1, ack
'2009-02-26 14:53:22:0643444', 2, ack
'2009-02-26 14:53:29:0806175', 3, ack
...
```

The mapping will be used to resolve the session id

```
[...,
 {session_id, {sid, 2}},
 ...]
```

# Aggregate

## etc\_ex\_Req.log

```
'2009-02-26 14:53:20:0000000',1  
'2009-02-26 14:53:20:0268786',2  
'2009-02-26 14:53:20:0531614',3  
'2009-02-26 14:53:21:0009257',4  
...
```

## etc\_ex\_ReqAck.log

```
'2009-02-26 14:53:29:0204458',1,ack  
'2009-02-26 14:53:22:0751754',3,ack  
'2009-02-26 14:53:29:0823714',4,ack  
...
```

## etc\_ex\_ReqErr.log

```
'2009-02-26 14:57:46:0474168',2,timeout  
...
```

## etc\_ex\_reqSMS.log

```
'2009-02-26 14:53:28:0701625',1,1,236  
'2009-02-26 14:53:22:0520858',3,2,160  
'2009-02-26 14:53:29:0529183',4,3,77  
...
```

## etc\_ex\_ackSMS.log

```
'2009-02-26 14:53:28:0870067',1,ack  
'2009-02-26 14:53:22:0643444',2,ack  
'2009-02-26 14:53:29:0806175',3,ack  
...
```

# Aggregate

## etc\_ex\_Req.log

```
'2009-02-26 14:53:20:0000000',1  
'2009-02-26 14:53:20:0268786',2  
'2009-02-26 14:53:20:0531614',3  
'2009-02-26 14:53:21:0009257',4  
...
```

## etc\_ex\_ReqAck.log

```
'2009-02-26 14:53:29:0204458',1,ack  
'2009-02-26 14:53:22:0751754',3,ack  
'2009-02-26 14:53:29:0823714',4,ack  
...
```

## etc\_ex\_ReqErr.log

```
'2009-02-26 14:57:46:0474168',2,timeout  
...
```

## etc\_ex\_reqSMS.log

```
'2009-02-26 14:53:28:0701625',1,1,236  
'2009-02-26 14:53:22:0520858',3,2,160  
'2009-02-26 14:53:29:0529183',4,3,77  
...
```

## etc\_ex\_ackSMS.log

```
'2009-02-26 14:53:28:0870067',1,ack  
'2009-02-26 14:53:22:0643444',2,ack  
'2009-02-26 14:53:29:0806175',3,ack  
...
```

# Aggregate

## etc\_ex\_Req.log

```
'2009-02-26 14:53:20:0000000',1  
'2009-02-26 14:53:20:0268786',2  
'2009-02-26 14:53:20:0531614',3  
'2009-02-26 14:53:21:0009257',4  
...
```

## etc\_ex\_ReqAck.log

```
'2009-02-26 14:53:29:0204458',1,ack  
'2009-02-26 14:53:22:0751754',3,ack  
'2009-02-26 14:53:29:0823714',4,ack  
...
```

## etc\_ex\_ReqErr.log

```
'2009-02-26 14:57:46:0474168',2,timeout  
...
```

## etc\_ex\_reqSMS.log

```
'2009-02-26 14:53:28:0701625',1,1,236  
'2009-02-26 14:53:22:0520858',3,2,160  
'2009-02-26 14:53:29:0529183',4,3,77  
...
```

## etc\_ex\_ackSMS.log

```
'2009-02-26 14:53:28:0870067',1,ack  
'2009-02-26 14:53:22:0643444',2,ack  
'2009-02-26 14:53:29:0806175',3,ack  
...
```

# Aggregate

## etc\_ex\_Req.log

```
'2009-02-26 14:53:20:0000000',1  
'2009-02-26 14:53:20:0268786',2  
'2009-02-26 14:53:20:0531614',3  
'2009-02-26 14:53:21:0009257',4  
...
```

## etc\_ex\_ReqAck.log

```
'2009-02-26 14:53:29:0204458',1,ack  
'2009-02-26 14:53:22:0751754',3,ack  
'2009-02-26 14:53:29:0823714',4,ack  
...
```

## etc\_ex\_ReqErr.log

```
'2009-02-26 14:57:46:0474168',2,timeout  
...
```

## etc\_ex\_reqSMS.log

```
'2009-02-26 14:53:28:0701625',1,1,236  
'2009-02-26 14:53:22:0520858',3,2,160  
'2009-02-26 14:53:29:0529183',4,3,77  
...
```

## etc\_ex\_ackSMS.log

```
'2009-02-26 14:53:28:0870067',1,ack  
'2009-02-26 14:53:22:0643444',2,ack  
'2009-02-26 14:53:29:0806175',3,ack  
...
```

# Abstract

```
[...,
 {sess_abstr,
   fun(Trs) ->
     AbstrFun = fun({Ts, {File}}) ->
       L = string:tokens(filename:basename(File), "_."),
       case lists:nth(length(L) - 1, L) of
         "Req" ->
           {Ts, req};
         "ReqSMS" ->
           {Ts, req_sms};
         "AckSMS" ->
           {Ts, ack_sms};
         "ReqAck" ->
           {Ts, req_ack};
         "ReqErr" ->
           {Ts, req_err}
       end,
     lists:map(AbstrFun, Trs)
   end},
 ...]
```



# Abstract

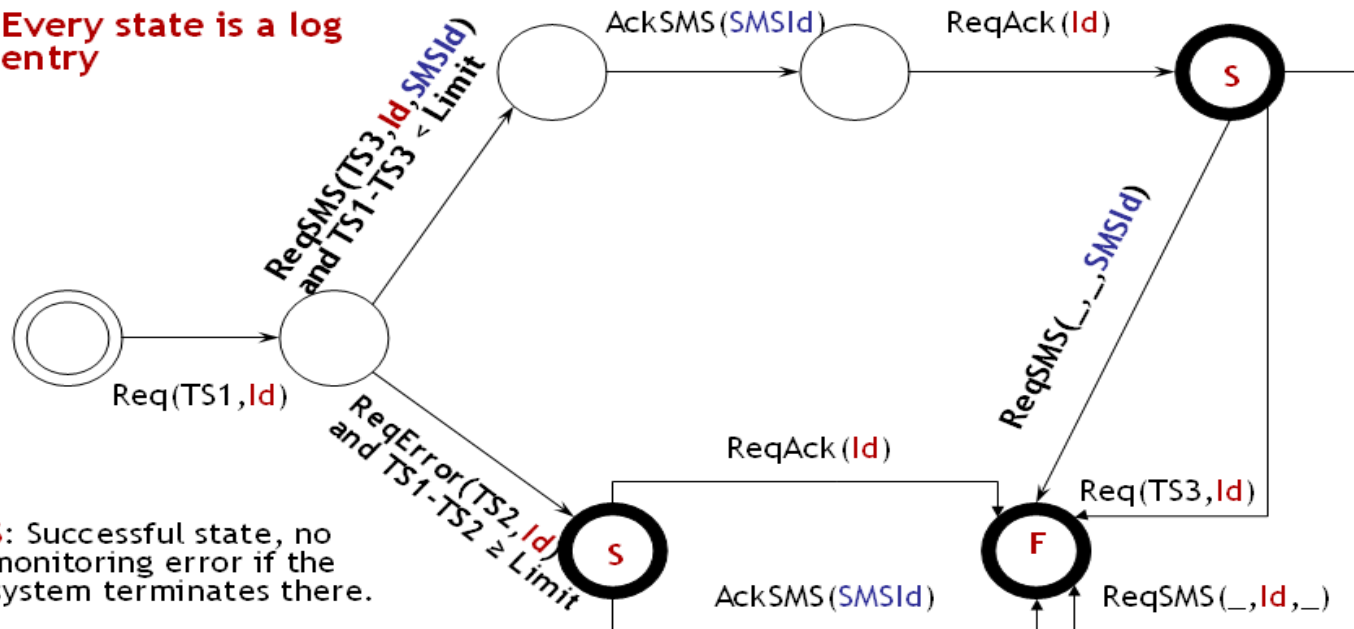
```
etc_ex Req.log
'2009-02-26 14:53:20:0000000',1
etc_ex ReqAck.log
'2009-02-26 14:53:29:0204458',1,ack
etc_ex_reqSMS.log
'2009-02-26 14:53:28:0701625',1,1,236
etc_ex_ackSMS.log
'2009-02-26 14:53:28:0870067',1,ack
```

- Parsed events are
  1. Aggregated
  2. Sorted by timestamps
  3. Passed to the abstraction function

```
{1,
 [{{2009,02,26},{14,53,20},0000000}, req},
  {{2009,02,26},{14,53,28},0701625}, req_sms},
  {{2009,02,26},{14,53,28},0870067}, ack_sms},
  {{2009,02,26},{14,53,29},0204458}, ack}}]
```

# Check

Every state is a log entry



**S:** Successful state, no monitoring error if the system terminates there.

**F:** Bad state, monitoring error if the system terminates there.

# Check

```
{statem,  
  [{states, [0,1,2,3,4,5,6]},  
   {trans,  [{0,1,req},  
             {1,2,req_sms, {lt,30}},  
             {2,3,ack_sms},  
             {3,4,req_ack},  
             {1,5,req_err, {geq,30}},  
             {5,6,req_ack},  
             {5,6,ack_sms},  
             {4,6,req},  
             {4,6,req_sms}]},  
   {terminal,[4,5,6]},  
   {good,    [4,5]}}  
}.
```

- State machine specified as a Labelled Transition System
- Time constraints for transitions in tenth of second
- Not all terminal states are “good” states – in this way we can expect specific classes of errors

# Check

No matching transition within time constraint  
for req\_sms in state 1

`{"1"}`

According to the specification, the SMS gateway should time out if it cannot connect to the SMSC within 3 seconds.

`{{{{2009,2,26},{14,53,20}},0},req}`

`{{{{2009,2,26},{14,53,28}},70162},req_sms}`

`{{{{2009,2,26},{14,53,28}},87006},ack_sms}`

`{{{{2009,2,26},{14,53,29}},20445},req_ack}`

# Interpreting the results

- What could this mean for the actual system?

No matching transition within time constraint.

- Error in the specification
- Error in the code
- No error in the code or in the specification

# Summary

- What Exago can be used for:
  - Revealing the cause of a fault
  - Detecting unnoticed errors
  - Testing the logging
  - Integration testing
  - Providing Quality Assurance

# Summary

- Advantages:
  - Language agnostic
  - Extensible interface
  - Open source
- Disadvantages:
  - Steep learning curve
  - Restrictions on log file semantics



# Summary

- Current work:
  - Interchangeable back-end
  - Inferring models from log files (Statechum)
- Future plans:
  - Improve scalability
  - Integration with other tools

# More information

Erlang Solutions website:  
<http://www.erlang-solutions.com>



ProTest Project website:  
<http://www.protest-project.eu>



Atilla Erdödi  
E-mail: [erdodi@elte.hu](mailto:erdodi@elte.hu)  
Twitter: @AtillaE



**Any questions?**