What is MINERful?

MINERful is a process mining, simulation, and analysis tool for declarative process specifications.

The main functionalities of MINERful are:

* Discovery of declarative specifications of process control-flows out of event logs. Event logs can be either real or synthetic, stored as XES, MXML, or text files (a collection of strings, in which every character is considered as an event, every line as a trace);
* Simulation of declarative specifications of process control-flows and creation of synthetic event logs. The input process specification can be given as a JSON file. The event logs can be exported as XES or MXML files;
* Simplification, and inconsistency removal, of declarative specifications of process control-flows.

Installation

As a requirement, a JRE 7+ should be installed on your machine.
To launch the `.sh` files, you have to run them on a Unix-based system with a BASH shell. Alternatively, the JAR files can be called directly, which may come handy if you are using a Windows system for instance.
No installation procedure is required.
This version has been tested on both a Ubuntu Linux (18.04) and a Mac OS X (Snow Leopard) machine.

Usage

You can run MINERful as a stand-alone software by invoking the command shell scripts or directly running the JAR with your Java installation. If you are a developer, you can include the JAR in your Java project and use use the MINERful APIs.

MINERful as a stand-alone application (with `.sh` files)

The easiest way to launch MINERful is to make use of the `.sh` files.

- To launch the miner: `run-MINERful.sh`
- To reduce the number of constraints in a process model: `run-MINERfulSimplifier.sh`
- To create synthetic logs: `run-MINERfulTracesMaker.sh`
- To run tests (as we did for the evaluation sections of our papers): `test-launchers/*.sh`

Each of those launchers can be invoked with the `-h` parameter. An explanation of all possible parameters that can be passed is written there (beware, they are quite an amount so take your time). Depending on the case, some parameters are specified in the `.sh` files, whereas others are left free.
In case, feel free to modify the constants declared at the beginning of the script, so as to customise them.

The scripts that end with the `-unstable.sh` suffix do not launch MINERful via its JAR, but directly using bytecode files. In this way, we can immediately try the modified source code without overwriting the JAR version.

Usage examples of MINERful as a stand-alone application (with `.sh` files)

- Mine an `XES` log file located in `/home/user/file.xes`:

```
run-MINERful.sh -iLF '/home/user/file.xes'
```

- Display the help screen:

```
run-MINERful.sh -h
```

Mine an `XES` log file located in `/home/user/file.xes` and export the discovered model in an `XML` file located in `/home/user/model-condec.xml`, formatted as Declare/ConDec. Set a support threshold of 0.95, a confidence level threshold of 0.25, and an interest factor threshold of 0.125 (if you have no idea what these parameters are about, check this paper out):

```
run-MINERful.sh -iLF '/home/user/file.xes' -condec '/home/user/model-condec.xml' -s
0.95 -c 0.25 -i 0.125
```

- Mine an `XES` log file located in `/home/user/file.xes`, with comprehensive debug lines

```
run-MINERful.sh -d all -iLF '/home/user/file.xes'
```

- Mine an `XES` log file located in `/home/user/file.xes`, with comprehensive debug lines. Let results be exported in a `CSV` file located in `/home/user/output.csv`:

```
run-MINERful.sh -d all -iLF '/home/user/file.xes' -CSV '/home/user/output.csv'
```

- Mine an `XES` log file located in `/home/user/file.xes`, with comprehensive debug lines. Set a support threshold of 0.95, a confidence level threshold of 0.25, and an interest factor threshold of 0.125 (if you have no idea what these parameters are about, check this paper out). Let results be exported in a `CSV` file located in `/home/user/output.csv`:

```
run-MINERful.sh -d all -iLF '/home/user/file.xes' -CSV '/home/user/output.csv'
-s 0.95 -c 0.25 -i 0.125
```

Usage examples of MINERful as a stand-alone application (running the JAR)

This is a little bit trickier, but necessary, in case you have a Microsoft Windows system.
From your prompt, type:

```
java -cp ".\lib\*;MINERful.jar" minerful.<LAUNCHER_CLASS> -h
```

where `LAUNCHER_CLASS` can be `MinerFulMinerStarter` to run the miner, `MinerFulTracesMakerStarter` to launch the builder of synthetic logs, `MinerFulSimplificationStarter` to use the simplification engine, and so on (kudos to Ralph A. for testing the commands on Windows).

The `-h` parameter appended at the end of the prompt shows and explains the parameters you can pass. They are exactly the same as the Linux/MacOS version.

MINERful APIs

For advanced users: You can use MINERful as a Java package and integrate it with your software! Check out the minerful.examples.api source code to see some examples.
In particular:

- discovery.MinerFulObserverInvokerOnXesFile demonstrates how to invoke the MINERful miner as an API, and subsequently observe the changes that are applied to the process model in a publish/subscribe fashion. Lastly, the model is saved as a Declare Map file.
- discovery.MinerFulCallerOnStringFile demonstrates how to call MINERful to discover a process model out of strings saved on a file.
- io.FromDeclareMapToJSONandXMLandCSV demonstrates how to call MINERful to load a process model stored as a Declare map XML file, and then store it converted into MINERful XML, CSV, and JSON formats.
- logmaking.FromCharactersProcessModelToLog demonstrates how to generate XES logs starting from the definitions of new constraints. It can be interesting also because it shows how to create a process model on the fly.
- logmaking.FromStringsProcessModelToLog demonstrates how to generate XES logs starting from the definitions of constraints exerted on activities identified by single characters (more or less the same as above).

A running example

Download from the 4TU datacenter website the real-world event log of the Business Process Intelligence Challenge (BPIC) 2012. Let us consider this log to be saved under the path `logs/BPIC2012/financial_log.xes.gz`.

To run MINERful, we locate our command shell in the working directory, and launch `run-MINERful.sh` as follows.

```
./run-MINERful.sh --in-log-file logs/BPIC2012/financial_log.xes.gz --in-log-evt-classifier 'logspec' --support 0.95 --confidence 0.5 --interest-factor 0.125 --prune-with 'hierarchyconflictredundancydouble' --save-as-csv models/bpic2012-declarative-model.csv --save-as-xml models/bpic2012-declarative-model.xml --save-as-json models/bpic2012-declarative-model.json
```

Alternatively, short names for options can be used:

```
./run-MINERful.sh -iLF logs/BPIC2012/financial_log.xes.gz -iLClassif 'logspec' -s 0.95 -c 0.5 -i 0.125 -prune 'hierarchyconflictredundancydouble' -oCSV models/bpic2012-declarative-model.csv -oXML models/bpic2012-declarative-model.xml -oJSON models/bpic2012-declarative-model.json
```

The rationale for those abbreviations is that the parameter prefixes pertain to the context: `-iL` for the `i`nput (event) `L`og, `-o…` for the `o`utput format in which discovered models are saved.

The *mandatory* `--in-log-file` (or `-iLF`) option locates the input event log. It should be in the XML-based XES format. The `--in-log-evt-classifier` (or `-iLClassif`) option dictates whether to consider as classifiers for events the sole activity names (default), or the pattern specified within the event log itself (in this case, the junction of activity name and lifecycle transition). The default thresholds for support, confidence level and interest factor of the returned constraints are by default 0.95, 0.25, and 0.125, respectively, but can be modified with the `--support`, `--confidence-level`, and `--interest-factor` option. More information on these metrics and how they are computed can be found in the paper at DOI:10.1145/2629447. Some post-processing can be required to polish the returned set of constraints from redundancies or possible internal inconsistencies (the lower the thresholds, the higher their amount). Here we select the most effective and time consuming strategy (`-prune`), that is to say,

the double-pass `hierarchyconflictredundancydouble`. The `--save-as-csv` (`-oCSV`), `save-as-xml` (`-oXML`), and `save-as-json` (`-oJSON`) options state where the CSV, XML, and JSON versions of the discovered constraints should be saved. The CSV file should show something like this:

Constraint	Template	Activation	Target	Support	Confidence level	Interest factor
AlternatePrecedence(A_PARTLYSUBMITTED+COMPLETE, A_ACCEPTED+COMPLETE)	AlternatePrecedence	A_ACCEPTED+COMPLETE	A_PARTLYSUBMITTED+COMPLETE	1	1	0.391
AlternatePrecedence(W_Completeren aanvraag+START, A_ACCEPTED+COMPLETE)	AlternatePrecedence	A_ACCEPTED+COMPLETE	W_Completeren aanvraag+START	1	0.563	0.22
AlternatePrecedence(A_PARTLYSUBMITTED+COMPLETE, A_ACTIVATED+COMPLETE)	AlternatePrecedence	A_ACTIVATED+COMPLETE	A_PARTLYSUBMITTED+COMPLETE	1	1	0.172
AlternatePrecedence(A_PARTLYSUBMITTED+COMPLETE, A_APPROVED+COMPLETE)	AlternatePrecedence	A_APPROVED+COMPLETE	A_PARTLYSUBMITTED+COMPLETE	1	1	0.172
AlternatePrecedence(A_PARTLYSUBMITTED+COMPLETE, A_CANCELLED+COMPLETE)	AlternatePrecedence	A_CANCELLED+COMPLETE	A_PARTLYSUBMITTED+COMPLETE	1	1	0.214
AtMostOne(A_DECLINED+COMPLETE)	AtMostOne	A_DECLINED+COMPLETE		1	0.583	0.34
NotSuccession(A_DECLINED+COMPLETE, W_Completeren aanvraag+START)	NotSuccession	A_DECLINED+COMPLETE	W_Completeren aanvraag+START	1	0.563	0.328
AlternatePrecedence(A_PARTLYSUBMITTED+COMPLETE, A_FINALIZED+COMPLETE)	AlternatePrecedence	A_FINALIZED+COMPLETE	A_PARTLYSUBMITTED+COMPLETE	1	1	0.383
AlternatePrecedence(W_Completeren aanvraag+START, A_FINALIZED+COMPLETE)	AlternatePrecedence	A_FINALIZED+COMPLETE	W_Completeren aanvraag+START	1	0.563	0.216
AtMostOne(A_PARTLYSUBMITTED+COMPLETE)	AtMostOne	A_PARTLYSUBMITTED+COMPLETE		1	1	1
Succession(A_PREACCEPTED+COMPLETE, W_Completeren aanvraag+COMPLETE)	Succession	A_PREACCEPTED+COMPLETE	W_Completeren aanvraag+COMPLETE	1	0.563	0.317

CoExistence(A_PREACCEPTED+COMPLETE, W_Completeren aanvraag+SCHEDULE)	CoExistence	A_PREACCEPTED+COMPLETE	W_Completeren aanvraag+SCHEDULE	1	0.563	0.317
ChainResponse(A_PREACCEPTED+COMPLETE, W_Completeren aanvraag+SCHEDULE)	ChainResponse	A_PREACCEPTED+COMPLETE	W_Completeren aanvraag+SCHEDULE	1	0.563	0.317
AlternatePrecedence(A_PARTLYSUBMITTED+COMPLETE, A_REGISTERED+COMPLETE)	AlternatePrecedence	A_REGISTERED+COMPLETE	A_PARTLYSUBMITTED+COMPLETE	1	1	0.172
Init(A_SUBMITTED+COMPLETE)	Init	A_SUBMITTED+COMPLETE		1	1	1
ChainSuccession(A_SUBMITTED+COMPLETE, A_PARTLYSUBMITTED+COMPLETE)	ChainSuccession	A_SUBMITTED+COMPLETE	A_PARTLYSUBMITTED+COMPLETE	1	1	1
AlternatePrecedence(A_PARTLYSUBMITTED+COMPLETE, O_ACCEPTED+COMPLETE)	AlternatePrecedence	O_ACCEPTED+COMPLETE	A_PARTLYSUBMITTED+COMPLETE	1	1	0.171
Precedence(W_Completeren aanvraag+START, O_CREATED+COMPLETE)	Precedence	O_CREATED+COMPLETE	W_Completeren aanvraag+START	1	0.563	0.216
Precedence(W_Completeren aanvraag+START, O_SELECTED+COMPLETE)	Precedence	O_SELECTED+COMPLETE	W_Completeren aanvraag+START	1	0.563	0.216
Precedence(W_Completeren aanvraag+START, O_SENT+COMPLETE)	Precedence	O_SENT+COMPLETE	W_Completeren aanvraag+START	1	0.563	0.216
Precedence(W_Completeren aanvraag+COMPLETE, O_SENT_BACK+COMPLETE)	Precedence	O_SENT_BACK+COMPLETE	W_Completeren aanvraag+COMPLETE	1	0.563	0.14
NotChainSuccession(W_Completeren aanvraag+COMPLETE, A_DECLINED+COMPLETE)	NotChainSuccession	W_Completeren aanvraag+COMPLETE	A_DECLINED+COMPLETE	1	0.563	0.328
Precedence(W_Completeren aanvraag+START, W_Completeren aanvraag+COMPLETE)	Precedence	W_Completeren aanvraag+COMPLETE	W_Completeren aanvraag+START	1	0.563	0.317

AtMostOne(W_Completeren aanvraag+SCHEDULE)	AtMostOne	W_Completeren aanvraag+SCHEDULE		1	0.563	0.317
NotSuccession(W_Completeren aanvraag+START, A_PREACCEPTED+COMPLETE)	NotSuccession	W_Completeren aanvraag+START	A_PREACCEPTED+COMPLETE	1	0.563	0.317
Response(W_Completeren aanvraag+START, W_Completeren aanvraag+COMPLETE)	Response	W_Completeren aanvraag+START	W_Completeren aanvraag+COMPLETE	1	0.563	0.317
Precedence(W_Completeren aanvraag+COMPLETE, W_Nabellen offertes+COMPLETE)	Precedence	W_Nabellen offertes+COMPLETE	W_Completeren aanvraag+COMPLETE	1	0.563	0.216
Precedence(W_Completeren aanvraag+START, W_Nabellen offertes+SCHEDULE)	Precedence	W_Nabellen offertes+SCHEDULE	W_Completeren aanvraag+START	1	0.563	0.216
Precedence(W_Completeren aanvraag+COMPLETE, W_Nabellen offertes+START)	Precedence	W_Nabellen offertes+START	W_Completeren aanvraag+COMPLETE	1	0.563	0.215
Precedence(W_Completeren aanvraag+COMPLETE, W_Valideren aanvraag+COMPLETE)	Precedence	W_Valideren aanvraag+COMPLETE	W_Completeren aanvraag+COMPLETE	1	0.563	0.138
Precedence(W_Completeren aanvraag+COMPLETE, W_Valideren aanvraag+SCHEDULE)	Precedence	W_Valideren aanvraag+SCHEDULE	W_Completeren aanvraag+COMPLETE	1	0.563	0.14
Precedence(W_Completeren aanvraag+COMPLETE, W_Valideren aanvraag+START)	Precedence	W_Valideren aanvraag+START	W_Completeren aanvraag+COMPLETE	1	0.563	0.138

The JSON version of the model would look like follows:


```
{
  "name": "Process model discovered out of financial_log.xes.gz",
  "tasks": ["W_Completeren aanvraag+COMPLETE", "W_Beoordelen fraude+START",
"A_PARTLYSUBMITTED+COMPLETE", "A_PREACCEPTED+COMPLETE", "W_Completeren
aanvraag+START", "O_SENT_BACK+COMPLETE", "O_SENT+COMPLETE", "A_ACTIVATED+COMPLETE",
"W_Nabellen incomplete dossiers+START", "O_CANCELLED+COMPLETE",
```

```
"A_SUBMITTED+COMPLETE", "O_SELECTED+COMPLETE", "W_Afhandelen leads+SCHEDULE",
"A_REGISTERED+COMPLETE", "W_Nabellen offertes+SCHEDULE", "W_Nabellen
offertes+COMPLETE", "A_DECLINED+COMPLETE", "W_Nabellen incomplete dossiers+SCHEDULE",
"W_Beoordelen fraude+COMPLETE", "W_Wijzigen contractgegevens+SCHEDULE",
"O_CREATED+COMPLETE", "W_Valideren aanvraag+SCHEDULE", "O_ACCEPTED+COMPLETE",
"W_Afhandelen leads+COMPLETE", "A_ACCEPTED+COMPLETE", "A_APPROVED+COMPLETE",
"A_CANCELLED+COMPLETE", "W_Valideren aanvraag+START", "W_Afhandelen leads+START",
"W_Valideren aanvraag+COMPLETE", "W_Nabellen offertes+START", "A_FINALIZED+COMPLETE",
"W_Beoordelen fraude+SCHEDULE", "W_Completeren aanvraag+SCHEDULE", "W_Nabellen
incomplete dossiers+COMPLETE", "O_DECLINED+COMPLETE"],
  "constraints": [{
      "template": "Precedence",
      "parameters": [
        ["W_Completeren aanvraag+COMPLETE"],
        ["W_Valideren aanvraag+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 0.5629250401161459,
      "interestFactor": 0.13803212758712555
    },
    {
      "template": "Precedence",
      "parameters": [
        ["W_Completeren aanvraag+START"],
        ["O_CREATED+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 0.5628486284098724,
      "interestFactor": 0.21568624371326583
    },
    {
      "template": "Precedence",
      "parameters": [
        ["W_Completeren aanvraag+COMPLETE"],
        ["W_Nabellen offertes+START"]
      ],
      "support": 1.0,
      "confidence": 0.5629250401161459,
      "interestFactor": 0.2153283984734031
    },
    {
      "template": "Precedence",
      "parameters": [
        ["W_Completeren aanvraag+START"],
        ["W_Nabellen offertes+SCHEDULE"]
      ],
      "support": 1.0,
      "confidence": 0.5628486284098724,
      "interestFactor": 0.21568624371326583
    },
    {
      "template": "NotSuccession",
      "parameters": [
        ["W_Completeren aanvraag+START"],
        ["A_PREACCEPTED+COMPLETE"]
```

```
    ],
    "support": 1.0,
    "confidence": 0.5628486284098724,
    "interestFactor": 0.31684158672694507
  },
  {
    "template": "AlternatePrecedence",
    "parameters": [
      ["A_PARTLYSUBMITTED+COMPLETE"],
      ["A_CANCELLED+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 1.0,
    "interestFactor": 0.21448765950943682
  },
  {
    "template": "AlternatePrecedence",
    "parameters": [
      ["W_Completeren aanvraag+START"],
      ["A_FINALIZED+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 0.5628486284098724,
    "interestFactor": 0.21568624371326583
  },
  {
    "template": "AtMostOne",
    "parameters": [
      ["W_Completeren aanvraag+SCHEDULE"]
    ],
    "support": 0.9996943531749064,
    "confidence": 0.5627529838648687,
    "interestFactor": 0.316787746017612
  },
  {
    "template": "Precedence",
    "parameters": [
      ["W_Completeren aanvraag+COMPLETE"],
      ["O_SENT_BACK+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 0.5629250401161459,
    "interestFactor": 0.13996776041399392
  },
  {
    "template": "NotChainSuccession",
    "parameters": [
      ["W_Completeren aanvraag+COMPLETE"],
      ["A_DECLINED+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 0.5629250401161459,
    "interestFactor": 0.32841236962533604
  },
  {
```

```
    "template": "Precedence",
    "parameters": [
      ["W_Completeren aanvraag+START"],
      ["W_Completeren aanvraag+COMPLETE"]
    ],
    "support": 0.9999582759627822,
    "confidence": 0.5628251440927526,
    "interestFactor": 0.31682836681678833
  },
  {
    "template": "CoExistence",
    "parameters": [
      ["A_PREACCEPTED+COMPLETE"],
      ["W_Completeren aanvraag+SCHEDULE"]
    ],
    "support": 1.0,
    "confidence": 0.5629250401161459,
    "interestFactor": 0.31688460078976444
  },
  {
    "template": "Init",
    "parameters": [
      ["A_SUBMITTED+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 1.0,
    "interestFactor": 1.0
  },
  {
    "template": "AlternatePrecedence",
    "parameters": [
      ["A_PARTLYSUBMITTED+COMPLETE"],
      ["A_ACTIVATED+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 1.0,
    "interestFactor": 0.17162069229005883
  },
  {
    "template": "AtMostOne",
    "parameters": [
      ["A_PARTLYSUBMITTED+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 1.0,
    "interestFactor": 1.0
  },
  {
    "template": "ChainResponse",
    "parameters": [
      ["A_PREACCEPTED+COMPLETE"],
      ["W_Completeren aanvraag+SCHEDULE"]
    ],
    "support": 1.0,
    "confidence": 0.5629250401161459,
```

```
    "interestFactor": 0.31688460078976444
  },
  {
    "template": "Precedence",
    "parameters": [
      ["W_Completeren aanvraag+COMPLETE"],
      ["W_Nabellen offertes+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 0.5629250401161459,
    "interestFactor": 0.21554346878749955
  },
  {
    "template": "AlternatePrecedence",
    "parameters": [
      ["A_PARTLYSUBMITTED+COMPLETE"],
      ["O_ACCEPTED+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 1.0,
    "interestFactor": 0.1713914571712386
  },
  {
    "template": "Precedence",
    "parameters": [
      ["W_Completeren aanvraag+START"],
      ["O_SELECTED+COMPLETE"]
    ],
    "support": 1.0,
    "confidence": 0.5628486284098724,
    "interestFactor": 0.21568624371326583
  },
  {
    "template": "NotSuccession",
    "parameters": [
      ["A_DECLINED+COMPLETE"],
      ["W_Completeren aanvraag+START"]
    ],
    "support": 1.0,
    "confidence": 0.5628486284098724,
    "interestFactor": 0.3283677907778234
  },
  {
    "template": "Precedence",
    "parameters": [
      ["W_Completeren aanvraag+COMPLETE"],
      ["W_Valideren aanvraag+START"]
    ],
    "support": 1.0,
    "confidence": 0.5629250401161459,
    "interestFactor": 0.13798911352430626
  },
  {
    "template": "ChainSuccession",
    "parameters": [
```

```
        ["A_SUBMITTED+COMPLETE"],
        ["A_PARTLYSUBMITTED+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 1.0,
      "interestFactor": 1.0
    },
    {
      "template": "AlternatePrecedence",
      "parameters": [
        ["A_PARTLYSUBMITTED+COMPLETE"],
        ["A_APPROVED+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 1.0,
      "interestFactor": 0.17162069229005883
    },
    {
      "template": "AlternatePrecedence",
      "parameters": [
        ["A_PARTLYSUBMITTED+COMPLETE"],
        ["A_FINALIZED+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 1.0,
      "interestFactor": 0.3832047069611064
    },
    {
      "template": "AlternatePrecedence",
      "parameters": [
        ["A_PARTLYSUBMITTED+COMPLETE"],
        ["A_REGISTERED+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 1.0,
      "interestFactor": 0.17162069229005883
    },
    {
      "template": "AtMostOne",
      "parameters": [
        ["A_DECLINED+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 0.5834033773974172,
      "interestFactor": 0.34035950075871324
    },
    {
      "template": "AlternatePrecedence",
      "parameters": [
        ["W_Completeren aanvraag+START"],
        ["A_ACCEPTED+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 0.5628486284098724,
      "interestFactor": 0.2199010496721691
```

```
    },
    {
      "template": "Response",
      "parameters": [
        ["W_Completeren aanvraag+START"],
        ["W_Completeren aanvraag+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 0.5628486284098724,
      "interestFactor": 0.31684158672694507
    },
    {
      "template": "AlternatePrecedence",
      "parameters": [
        ["A_PARTLYSUBMITTED+COMPLETE"],
        ["A_ACCEPTED+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 1.0,
      "interestFactor": 0.3906930541758997
    },
    {
      "template": "Precedence",
      "parameters": [
        ["W_Completeren aanvraag+COMPLETE"],
        ["W_Valideren aanvraag+SCHEDULE"]
      ],
      "support": 1.0,
      "confidence": 0.5629250401161459,
      "interestFactor": 0.13996776041399392
    },
    {
      "template": "Succession",
      "parameters": [
        ["A_PREACCEPTED+COMPLETE"],
        ["W_Completeren aanvraag+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 0.5629250401161459,
      "interestFactor": 0.31688460078976444
    },
    {
      "template": "Precedence",
      "parameters": [
        ["W_Completeren aanvraag+START"],
        ["O_SENT+COMPLETE"]
      ],
      "support": 1.0,
      "confidence": 0.5628486284098724,
      "interestFactor": 0.21568624371326583
    }
  ]
}
```

Considerations

To speed up the post-processing, other techniques than `hierarchyconflictredundancydouble` (say, `hierarchyconflictredundancy`, or just `hierarchy`, the default) can be used. However, this implies less accuracy in the detection (and removal) of the redundant constraints. Generally, the mining per se is very fast. The post-processing takes most of the computation time.

To filter out constraints, the thresholds for support, confidence level and interest factor can be increased. Mind however that the confidence level, and even more the interest factor, are rather sensitive to small changes (so, a slight increase may induce many constraints to be cut out). In this running example we kept thresholds quite low so as to show a number of different constraints.

Using MINERful to create synthetic event logs

MINERful can be used also to create event logs that comply with a given declarative workflow specification.
There are two ways to do that: either via command line, or using MINERful as a Java package to be integrated in your software project.

Creating event logs via command line

Suppose you have a declarative process specification as follows:

```
{ name: "Scientific paper evaluation process",
  constraints: [
                           {template: "respondedexistence", parameters: [["Submit
abstract"],["Write new paper"]]},
                           {template: "response", parameters: [["Submit
paper"],["Send confirmation email"]]},
                           {template: "succession", parameters: [["Submit
paper"],["Review paper"]]},
                           {template: "precedence", parameters: [["Review
paper"],["Accept paper"]]},
                           {template: "notsuccession", parameters: [["Reject
paper"],["Submit paper"]]},
                           {template: "notcoexistence", parameters: [["Accept
paper"],["Reject paper"]]}
] }
```

Let us assume this file is saved in `/home/claudio/Desktop/exampleprocess.json`.
To generate an event log consisting of 2000 traces, the length of whose traces is between 2 and 24 tasks, and store it in XES format in `/home/claudio/Desktop/examplelog.xes`, run the following command

```
run-MINERfulEventLogMaker.sh --input-model-file
'/home/claudio/Desktop/exampleprocess.json' --input-model-encoding json  --size 2000
--minlen 2 --maxlen 24 --out-log-encoding xes --out-log-file
'/home/claudio/Desktop/examplelog.xes'
```

The same result can be achieved with a shorthand notation for passing parameters:

```
run-MINERfulEventLogMaker.sh -iMF '/home/claudio/Desktop/exampleprocess.json' -iME
json -oLL 2000 -oLm 2 -oLM 24 -oLE xes -oLF '/home/claudio/Desktop/examplelog.xes'
```

The rationale for those abbreviations is that the parameter prefixes pertain to the context: `-iM` for the `i`nput `M`odel, `-oL` for the `o`utput (event) `L`og.
As usual, to have an overview of the parameters, their meaning and their default, you can run:

```
run-MINERfulEventLogMaker.sh -h
```

Indeed one can use other formats for the input process model (for instance, the native MINERful XML format, just out of a previous mining task) or the output log (for instance, the MXML old process mining input format).
Because the event log is for all means and purposes a full-fledged XES event log, you can run the discovery algorithm of MINERful to see what model comes out. This turns out to be very useful for testing purposes:

```
run-MINERful.sh -iLF '/home/claudio/Desktop/examplelog.xes'
```

Creating event logs via API

- logmaking.FromDeclareMapToLog demonstrates how to generate XES logs from an existing Declare map XML file. It is worth having a look at it to see how to convert process models from the Declare Maps Miner format, for those who used that tool in ProM.
- logmaking.FromJsonProcessModelToLog demonstrates how to generate XES logs starting with the definitions of constraints specified with JSON objects.
- simplification.MinerFulSimplificationInvokerOnDeclareMapFile demonstrates how to load a Declare Map file as a process model, then run the simplification engine of MINERful to remove the redundant constraints.

Publications

For more information on how the log generator works, or how it has been used, feel free to refer to the following publications:

- Simulation of declarative models:

  Claudio Di Ciccio, Mario Luca Bernardi, Marta Cimitile, Fabrizio Maria Maggi: Generating Event Logs Through the Simulation of Declare Models. EOMAS@CAiSE 2015: 20-36

  - DOI: 10.1007/978-3-319-24626-0_2
  - Pre-print: diciccio.net/claudio/preprints/DiCiccio-etal-EOMAS2015.pdf

- Testing the sensitivity of constraints to noise:

  Claudio Di Ciccio, Massimo Mecella, Jan Mendling: The Effect of Noise on Mined Declarative Constraints. SIMPDA 2015: 1-24

  - DOI: 10.1007/978-3-662-46436-6_1
  - Pre-print: diciccio.net/claudio/preprints/DiCiccio-etal-SIMPDApp2015-EffectofNoise.pdf

Main publications

Selected publications about MINERful and presentation slides:

- The main discovery algorithm:

  Claudio Di Ciccio, Massimo Mecella: On the Discovery of Declarative Control Flows for Artful Processes. ACM Trans. Management Inf. Syst. 5(4): 24:1-24:37 (2015)

  - DOI: 10.1145/2629447
  - Presentation: https://www.slideshare.net/cdc08x/automated-discovery-of-declarative-process-models

- Discovery of more target-branched (read: more complex) declarative models:

  Claudio Di Ciccio, Fabrizio Maria Maggi, Jan Mendling: Efficient discovery of Target-Branched Declare constraints. Inf. Syst. 56: 258-283 (2016)

  - DOI: 10.1016/j.is.2015.06.009

- Getting rid of redundancies and inconsistencies:

  Claudio Di Ciccio, Fabrizio Maria Maggi, Marco Montali, Jan Mendling: Resolving inconsistencies and redundancies in declarative process models. Inf. Syst. 64: 425-446 (2017)

  - DOI: 10.1016/j.is.2016.09.005
  - Presentation: https://www.slideshare.net/cdc08x/resolving-inconsistencies-and-redundancies-in-declarative-process-models

- Retaining only non-vacuously satisfied (read: relevant) constraints:

  Claudio Di Ciccio, Fabrizio Maria Maggi, Marco Montali, JanMendling: On the relevance of a business constraint to an event log. Inf. Syst. 78: 144-161 (2018)

  - DOI: 10.1016/j.is.2018.01.011
  - Presentation: https://www.slideshare.net/cdc08x/semantical-vacuity-detection-in-declarative-process-mining

- Simulation of declarative models:

  Claudio Di Ciccio, Mario Luca Bernardi, Marta Cimitile, Fabrizio Maria Maggi: Generating Event Logs Through the Simulation of Declare Models. EOMAS@CAiSE 2015: 20-36

  - DOI: 10.1007/978-3-319-24626-0_2

Other software packages using MINERful

A ProM package of MINERful exists, although it is in beta version and with limited functionalities: It is available in the ProM Nightly Build SVN repository. More information is available in this paper:

- Claudio Di Ciccio, Mitchel H. M. Schouten, Massimiliano de Leoni, Jan Mendling: Declarative Process Discovery with MINERful in ProM. BPM (Demos) 2015: 60-64. URL: http://ceur-ws.org/Vol-1418/paper13.pdf

A GUI-equipped log generator is also in its beta version, based on the Declare Designer tool: It is available for download on the Synthetic log generator GitHub page. More information is available in this paper:

- Claudio Di Ciccio, Mario Luca Bernardi, Marta Cimitile, Fabrizio Maria Maggi: Generating Event Logs Through the Simulation of Declare Models. EOMAS@CAiSE 2015: 20-36. DOI: 10.1007/978-3-319-24626-0_2

License

Please read the LICENSE file published in the MINERful GitHub repo.

Contact

Please contact the developer, Claudio Di Ciccio, for any information, comment or bug reporting: dc.claudio@gmail.com.