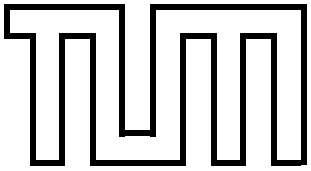

FAKULTÄT FÜR INFORMATIK
Technische Universität München

Specification Draft

THE XMTS FILE FORMAT

AUTHOR: SALOMON SICKERT
DATE: JUNE 15, 2013

1 OVERVIEW

1.1 INTRODUCTION

Naturally, one needs a standard file format for specifying, storing, and exchanging modal transition systems. In the closely related graph file formats it is either impossible or very tedious to specify properties, such as distinct must- and may-transitions or obligation functions. Hence a new file format was designed, which is loosely inspired by the *GML* format, a simple plain-text graph file format [Him].

1.2 BASIC CONCEPT

The *xmts* file format is stored as plain text and based on the concept of lists - also called *records* - consisting of key-value pairs. This approach makes it very easy to extend and to adapt it. Currently the format specifies keywords to express the following MTS extensions: DMTS, OTS, BMTS, PMTS, MTSD, MTSD+DPS. The last two extensions were introduced in [Ben+12].

An example can be found in Listing 1.1. In the first line, it is stated that the specified system is a PMTS with the id (`param1`). The attached record - delimited by **[** and **]** - contains the optional description as the first entry. Furthermore we specify with the **parameter** and **action** keywords the supported parameters and actions of the system. To make the format easier to read and write, outgoing transitions and the obligation function are directly listed in the record of the state. The obligation is given as a boolean formula - the exact production rules of which can be found in the next section. The **all** keyword is a short hand for the conjunction of all transitions. The scheme - *action* **->** *successorState* - used for specifying transitions intentionally diverges from the strict key-value design for reasons of convenience.

Listing 1.1: xmts representation of a system

```
pmts param1 [
    label "Abstract Traffic Light System"
    parameter reqY

    action go
    action stop
    action ready

    state green [
        obligation ([stop, red] <-> ![ready, yellow])
            & (reqY <-> [ready, yellow])
        ready -> yellow
        stop -> red
    ]

    state yellow [
        obligation all
        stop -> red
    ]

    state red [
        obligation ([go, green] <-> ![ready, yellowRed])
            & (reqY <-> [ready, yellowRed])
        ready -> yellowRed
        go -> green
    ]

    state yellowRed [
        obligation all
        go -> green
    ]
]
```

2 FORMAL LANGUAGE DEFINITION

2.1 ABSTRACT GRAMMAR

ADDITIONAL NOTES

- This version of the grammar and the antlr3 grammar is not identical due to technical reasons.
- To improve readability white space is not explicitly noted, however needed between keywords and values.

TOPLEVEL RULES

⟨*xmts*⟩ ::= ⟨*mts*⟩ | ⟨*dmts*⟩ | ⟨*bmts*⟩ |⟨*pmts*⟩ | ⟨*mtsd*⟩ | ⟨*mtsd+dps*⟩

⟨*mts*⟩ ::= **mts** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*initial*⟩? ⟨*action*⟩* ⟨*state*⟩+ **]**

⟨*dmts*⟩ ::= **dmts** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*initial*⟩? ⟨*action*⟩* ⟨*state*⟩+ **]**

⟨*bmts*⟩ ::= **bmts** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*initial*⟩? ⟨*action*⟩* ⟨*state*⟩+ **]**

⟨*pmts*⟩ ::= **pmts** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*initial*⟩? ⟨*parameter*⟩* ⟨*action*⟩* ⟨*state*⟩+ **]**

⟨*mtsd*⟩ ::= **mtsd** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*initial*⟩ ⟨*action*⟩* ⟨*state*⟩+ **]**

⟨*mtsd+dps*⟩ ::= **mtsd+dps** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*initial*⟩ ⟨*parameter*⟩* ⟨*action*⟩* ⟨*state*⟩+ **]**

COMMON PROPERTIES

⟨*label*⟩ ::= **label** ⟨*string*⟩

⟨*id*⟩ ::= ⟨*char*⟩ (⟨*char*⟩ | ⟨*digit*⟩ | _)*

ACTION PROPERTIES

⟨*action*⟩ ::= **action** ⟨*id*⟩
| **action** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*requirement*⟩? ⟨*cost*⟩? **]**

⟨*requirement*⟩ ::= **requirement** ⟨*formula*⟩

⟨*cost*⟩ ::= **running_cost** ⟨*number*⟩
| **running_cost** ⟨*signed_number*⟩

PARAMETER PROPERTIES

⟨*parameter*⟩ ::= **parameter** ⟨*id*⟩
| **parameter** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*investment*⟩? **]**

⟨*investment*⟩ ::= **investment_cost** ⟨*number*⟩

STATE PROPERTIES

⟨*state*⟩ ::= **state** ⟨*id*⟩
| **state** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*obligation*⟩? ⟨*position*⟩? ⟨*transition*⟩* **]**

⟨*obligation*⟩ ::= **obligation** ⟨*formula*⟩
| **obligation all**

⟨*position*⟩ ::= **position (** ⟨*number*⟩ **,** ⟨*number*⟩ **)**

⟨*transition*⟩ ::= ⟨*id*⟩ **->** ⟨*id*⟩
| ⟨*id*⟩ **->** ⟨*id*⟩ **[** ⟨*label*⟩? ⟨*duration*⟩? **]**

⟨*duration*⟩ ::= **duration (** ⟨*number*⟩ **,** ⟨*number*⟩ **)**
| **duration [** ⟨*number*⟩ **,** ⟨*number*⟩ **]**

⟨*initial*⟩ ::= **initial_state** ⟨*id*⟩

BOOLEAN FORMULA

⟨*formula*⟩ ::= ⟨*unit*⟩
| **!** ⟨*unit*⟩
| ⟨*unit*⟩ **&** ⟨*unit*⟩ (**&** ⟨*unit*⟩)*
| ⟨*unit*⟩ **|** ⟨*unit*⟩ (**|** ⟨*unit*⟩)*
| ⟨*unit*⟩ **->** ⟨*unit*⟩
| ⟨*unit*⟩ **<->** ⟨*unit*⟩

⟨*unit*⟩ ::= **true** | **false** | ⟨*id*⟩ | ⟨*id*⟩ **.** ⟨*id*⟩ | **(** ⟨*formula*⟩ **)**

BASIC RULES

⟨*char*⟩ ::= **a** | ... | **z** | **A** | ... | **Z**

⟨*digit*⟩ ::= **0** | ... | **9**

⟨*number*⟩ ::= (**1** | ... | **9**) ⟨*digit*⟩*
| **0**

⟨*signed_number*⟩ ::= (**+** | **-**) ⟨*number*⟩

⟨*string*⟩ ::= **"** Σ * **"** *

2.2 KEYWORD REFERENCE

ADDITIONAL NOTES

- The keywords are grouped by context.
- The third column specifies if the keyword must have an attached record (✓), can have one (𝒪), or must not have one (×).

* This production rule is greedy

Context	Keyword	Rec.	Meaning
	mts	✓	<*value*> is an MTS
	mtsd	✓	<*value*> is an MTSD
	mtsd+dps	✓	<*value*> is an MTSD with dual-priced scheme
	pmts	✓	<*value*> is a PMTS
All	**label**	×	<*value*> is the description of the record
System	**initial_state**	×	<*value*> is the initial state
	parameter	$\mathcal{O}$	Add <*value*> to P
	action	$\mathcal{O}$	Add <*value*> to Σ
	state	$\mathcal{O}$	Add <*value*> to S
Action	**requirement**	×	<*value*> is the requirement of the action
	running_cost	×	<*value*> is the running cost of the action
Parameter	**investment_cost**	×	<*value*> is the investment cost of the parameter
State	**obligation**	×	<*value*> is the obligation function of the state. Missing obligations will be interpreted as **all**
	position	×	Hint for placing the state in a graphical representation.
	->	$\mathcal{O}$	Add (<*value1*>, <value2>) to $T(s)$
Transition	**duration**	×	<*value1*> is the duration interval of the transition

Table 2.1: Keyword Reference

2.3 SEMANTICALLY WRONG INPUT

Although there exist safeguards in the grammar to avoid semantically wrong input, e.g. parameter specification after claiming the system is an MTS, invalid descriptions are still possible. In this case the behaviour is undefined, and it is up to the implementation if invalid information is discarded or the input is completely rejected.

[Ben+12] Nikola Beneš et al. "Dual-Priced Modal Transition Systems with Time Durations". In: *Logic for Programming, Artificial Intelligence, and Reasoning*. Ed. by Nikolaj Bjørner and Andrei Voronkov. Vol. 7180. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 122–137. ISBN: 978-3-642-28716-9. DOI: 10.1007/978-3-642-28717-6_12. URL: http://dx.doi.org/10.1007/978-3-642-28717-6_12.

[Him] Michael Himsolt. *GML: A portable Graph File Format*. Tech. rep. Universität Passau. URL: www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-technical-report.pdf.