

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Softwarový nástroj pro konfigurování distribuovaných
simulačních modelů využívajících webovou simulaci

Příručka vývojáře appletu

Bc. Štěpán Karták

Součást diplomové práce
2013

Anotace

Tento dokument podrobně popisuje požadavky na applet použitelný v rámci aplikace. Je zde podrobně rozepsán formát a význam zpráv pro komunikaci s dodávaným serverem, dále pak podrobný popis složení konfiguračního XML a další informace.

Pro verzi aplikace

20130501-1800

Obsah

Seznam obrázků	4
Seznam tabulek	4
Seznam zkratk	5
1 Obecné informace	6
2 Komunikační zprávy	7
2.1 Serverová část	8
2.2 Serverová část / Použití a chování serveru	9
2.3 Klientská část, použití a chování	10
2.4 Chybové stavy.....	11
3 Konfigurační soubor XML appletu	13
4 Konfigurační soubor XML simulace	19
5 Předávané parametry appletu	24
6 Komunikace Administrace se serverem	26
7 Názvy modelů	28
Příloha A – XML Schema appletu	29
Příloha B – XML Schema simulace	31

SEZNAM OBRÁZKŮ

Obrázek 1 – Schéma komunikace klient-server-klient	7
--	---

SEZNAM TABULEK

Tabulka 1 – Přehled chybových stavů	11
---	----

SEZNAM ZKRATEK

ADT	Abstraktní datový typ
AJAX	Asynchronous JavaScript and XML
CSS	Cascading Style Sheets
DOM	Document Object Model
JNLP	Java Network Launch Protocol
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
LVT	Local virtual time, lokální virtuální čas
LCC	Local Causality Constraint, podmínka lokální kauzality
OOP	Object-oriented programming, objektově orientované programování
PHP	PHP: Hypertext Preprocessor, serverový programovací jazyk
UML	Unified Modeling Language
(X)HTML	(Extensible) HyperText Markup Language
XML	Extensible Markup Language

1 OBECNÉ INFORMACE

Tento dokument pojednává technických detailech a požadavcích, který jsou kladeny na applety v rámci aplikace. O koncepci celé aplikace a konkrétních použitých technikách synchronizace simulace pojednává základní dokument – „KartakS_SoftwarovyNastroj_JH_2013.pdf“ přiložený na CD.

XML XPath

Dvěma lomítky vždy začíná cesta podle specifikace XML XPath, pokud není uvedeno jinak.

Balíčky a segmenty programového kódu

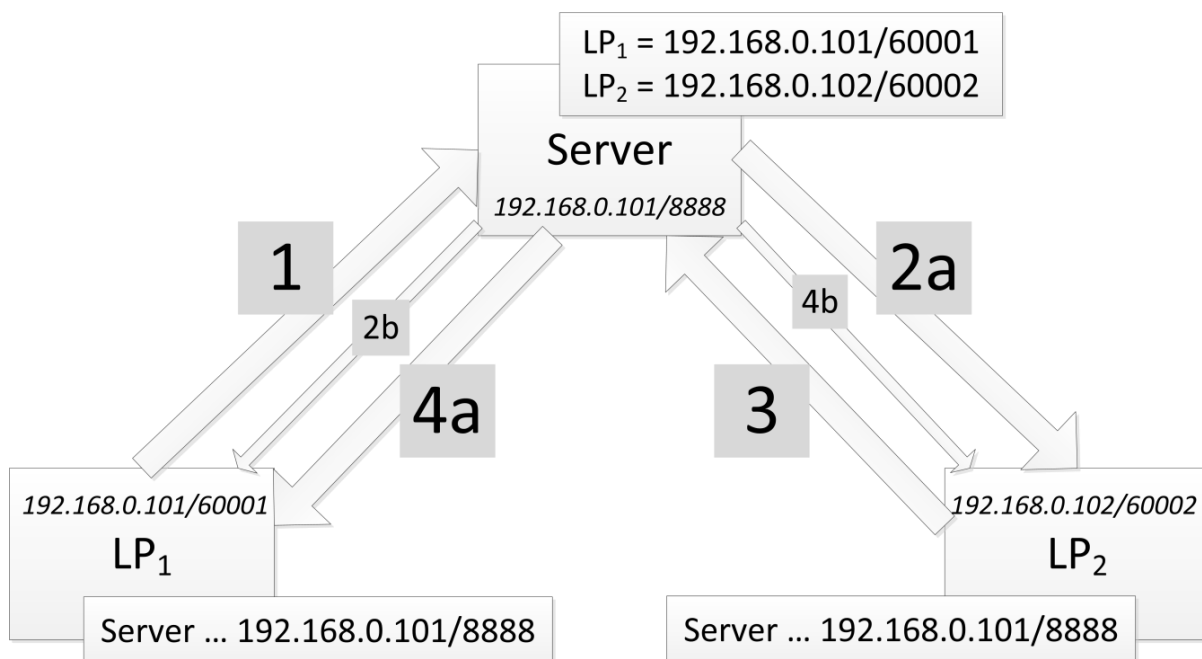
Je odkazováno na výchozí dodávané zdrojovou soubory serveru a ukázkových modelu. Vše v jazyce Java, pokud není uvedeno jinak.

Termíny

- **Administrace** – zkrácený název samotného webového nástroje pro konfigurování distribuovaných simulačních modelů.
- **Model** – interně užívaný název pro logický proces v rámci celého distribuovaného simulačního modelu.

2 KOMUNIKAČNÍ ZPRÁVY

Schéma komunikace klienti-server-klient zobrazeno na obrázek 1.



Obrázek 1 – Schéma komunikace klient-server-klient

Zprávy mají formát:

```
sender=LP1&client.name=LP2&req=client/&udalost=prijezd&typ=vozidlo&cislo=23  
&time.u=1365714014599&sender=LP1&time=1365714011599
```

Jednotlivé datové položky zpráva jsou tvořeny páry „klíč=hodnota“ oddělené znakem ampersand „&“. Časová razítka odpovídají číselnému formátu typu *long*¹ – podle epochy UNIXu v milisekundách v desítkové soustavě uložené ve formě řetězce.

Zpráva je rozdělena na dvě části znakem lomítka „/“:

- **serverová** část na levé straně, obsahuje informace o odesílateli, cílovém klientu a typ požadavku (v našem případě *client* – zpráva bude přeposlána odesílateli),
- **klientská** část na pravé straně s informacemi o události (*udalost*), časovém razítku události (*time.u*) případně dalšími parametry (*typ*, *cislo*) týkající se události či odesílatele (*sender*)².

¹ Běžně dostupný datový typ používaný v programovacích jazycích. Obecně se jedná o 64bitové celé číslo.

² Položek je více, většinou se jedná o dodatečná data čistě informativního charakteru, jako například položka *time*, která obsahuje LVT odesílatele. Konkrétně položka *time* má význam čistě pro ladění programu.

Pokud není uvedeno jinak, hodnoty jsou textové informace. Hodnoty atributů jsou předávány pomocí interní funkce Javy:

```
String save = URLEncoder.encode("Původní řetězec".toString(), "UTF-8")
// převod do bezpečného řetězce

String puvodniRetetec = URLEncoder.decode(save.toString(), "UTF-8")
// převod zpět do běžného / původního řetězce
```

2.1 Serverová část

req [Známa konstanta `Attributes.TYPE_SERVER_REQUEST`]

Specifikuje typ zprávy/požadavku, platní hodnoty:

- **response**
[`Attributes.TYPE_SERVER_REQUEST_VAL_RESPONSE`]
Zpráva je odpověď klienta jinému klientovi.
- **sresponse**
[`Attributes.TYPE_SERVER_REQUEST_VAL_SRESPONSE`]
Zpráva je odpověď serveru zaslaná klientovi o zpracování požadavku.
- **register**
[`Attributes.TYPE_SERVER_REQUEST_VAL_REGISTER`]
Zpráva je odpověď klienta jinému klientovi. Server ji přeposílá.
- **client**
[`Attributes.TYPE_SERVER_REQUEST_VAL_CLIENT`]
Zpráva je adresována klientovi a server ji přeposílá.
- **hello**
[`Attributes.TYPE_SERVER_REQUEST_VAL_HELLO`]
Zpráva informuje server o novém klientovi.
- **stop**
[`Attributes.TYPE_SERVER_REQUEST_VAL_STOP`]
Zpráva ukončí činnost serveru – vypne ho.

client.name [`Attributes.TYPE_SERVER_CLIENT_NAME`]

Specifikace jména klienta, pro kterého je zpráva určena.

state [Attributes.**TYPE_STATE**]

Chybový stav. Používá se výlučně v odpovědích serveru na požadavek.

Tato hodnota je parsována na celé číslo. Hodnota 0 značí stav „vše v pořádku“. Ostatní stavy viz tabulka 1.

message [Attributes.**TYPE_MESSAGE**]

Textová informace o požadavku. Má význam zpravidla při hodnotě **state**<>0, kdy v atributu **message** nalezneme dodatečnou textovou informaci.

sender [Attributes. **TYPE_SENDER**]

Název klienta odesílatele zprávy.

key [Attributes.**TYPE_SERVER_KEY**]

Klient zasílá atribut **key** jako jednoznačný identifikátor zprávy. Server tento „klíč“ zopakuje v odpovědi.

2.2 Serverová část / Použití a chování serveru

Server standardně odpovídá zprávou, která obsahuje pouze serverovou část, a to v této sestavě:

```
key, request=sresponse, state, message
```

req=response

Pokud server obdrží požadavek (**req**) s hodnotu **response**, pak se zachová podle toho, zda je uveden atribut **client.name**. Pokud je uveden **client.name**, pak je klientská část zprávy přeposlána specifikovanému klientovi. Pokud **client.name** není uvedeno, zpráva se nepřeposílá.

Klientovi, který tento požadavek poslal, je odeslána zpráva o tom, zda se přeposlání zdařilo.

req=hello

Tento požadavek vyžaduje další atribut **port**, který specifikuje číslo portu, na kterém klient poslouchá. Hodnota je parsována na celé číslo. Dále je požadován atribut **client.name**, který je vyžadován. Tento atribut bude použit jako název klienta v další komunikaci.

Klientovi, který tento požadavek poslal, je odeslána zpráva o tom, zda se ho podařilo na serveru zaregistrovat.

req=sresponse

Tento stav by se neměl nikdy vyskytnout.

req=client, req=register

Přepošle zprávu klientovi, specifikovaném v atributu **client.name**, který je vyžadován.

Klientovi, který tento požadavek poslal, je odeslána zpráva o tom, zda se ho podařilo zprávu zaregistrovat.

req=stop

Zastaví server.

Klientovi, který tento požadavek poslal, je odeslána zpráva o tom, zda byl požadavek úspěšně zpracován a server bude ukončen.

2.3 Klientská část, použití a chování

message [Attributes.**TYPE_MESSAGE**]

Textová zpráva.

service [Attributes.**TYPE_SERVICE**]

Celočíselná hodnota. Platné jsou „0“ – toto není *service* zpráva a „1“ – toto je *service* zpráva.

sender [Attributes.**TYPE_SENDER**]

Název klienta odesílatele zprávy.

cinnost [Attributes.**TYPE_NAME_CINNOST**]

Název činnosti, kde bude vykonána předávaná událost. Tento atribut je často povinný.

time.u [Attributes.**TYPE_TIME_UDALOSTI**]

Hodnota typu *long*, je to časové razítko dle epochy UNIXu. Vyžadováno u zpráv s uvedeným **cinnost** a **service**<>1.

time [Attributes.**TYPE_MY_TIME**]

LVT klienta. Má pouze informativní charakter a není povinný.

Další libovolné atributy dle událostí.

2.4 Chybové stavy

Server vrací celou řadu stavů – viz tabulka 1. Ne všechny stavy jsou implementovány.

Tabulka 1 – Přehled chybových stavů

Hodnota	Konstanta (Balíček NetworkMessage)	Význam	Zasílá server	Zasílá klient
0	.STATE_OK	Vše v pořádku	Ano	Ano
1	.STATE_SERVER_IS_NOT_AVAILABLE	Server není dostupný	Ano	Ano
10	.STATE_CLIENT_UNKNOWN	Požadujete neznámého klienta	Ano	Ne
11	.STATE_CLIENT_IS_NOT_AVAILABLE	Klient je známý, ale nepodařilo se zprostředkovat přednost zprávy	Ano	Ne
12	.STATE_CLIENT_CALENDAR_ERROR	Do fronty klienta, se nepodařilo zařadit událost	Ne	Ano
100	.STATE_ERROR_BAD_REQUEST	Špatný požadavek (req)	Ano	Ano
101	.STATE_ERROR_UNKNOWN_REQUEST	Neznámý požadavek (req)	Ano	Ano
102	.STATE_ERROR_UNKNOWN_NAME_CINNOST	Neznámá činnost (cinnost)	Ano	Ano
999	.STATE_ERROR_UNKNOWN	Neznámá chyba	Ano	Ano
-1	.STATE_ERROR_ANY	Stav se nepodařilo určit	Ano	Ano

Každý stav (**state**), který je přenášen v serverové části, je doprovázen zprávou (**message**) v serverové části.

Dodatečné informace v klientské části pro chyby:

- NetworkMessage.**STATE_CLIENT_CALENDAR_ERROR**

Přidá do klientské části:

- **calendar.log.level** [Calendar.**ATTR_CALENDAR_LOG_LEVEL**]

Úroveň chyby loggeru.

- **calendar.time** [Calendar.ATTR_CALENDAR_TIME]
Aktuální čas LVT.
- **calendar.message** [Calendar.ATTR_CALENDAR_MESSAGE]
Detailní zpráva.

3 KONFIGURAČNÍ SOUBOR XML APPLETU

XML Schema je k dispozici v příloze A. Ve schéma jsou základní informace uvedené v části `<xs:annotation>`, v této kapitole budou rozebrány podrobněji. Rozšířené datové typy a strukturované elementy jsou podrobně rozepsány.

Root element `<model>` (`//model`)

Obsahuje podelementy:

- **`<name>` Název appletu**
[umístění: `//model/name`, datový typ: `string-long`, povinný]
- **`<description>` Vyčerpávající popis appletu**, měl by obsahovat obecný popis, podmínky použití, význam a způsob užití, případně specifiky chování, atp.
[umístění: `//model/description`, datový typ: `xs:string` (doporučeno použít HTML formátování), povinný]
- **`<visual>` Informace pro vizualizaci appletu v návrháři**
[umístění: `//model/visual`, datový typ: rodičovský element (podelementy uvedeny dále v textu), povinný]
- **`<settings>` Specifikuje nastavení modelu**
[umístění: `//model/settings`, datový typ: rodičovský element, povinný]
- **`<inputs>` Vstupní body modelu**
[umístění: `//model/inputs`, datový typ: rodičovský element, povinný]
- **`<outputs>` Výstupní body modelu**
[umístění: `//model/inputs`, datový typ: rodičovský element, povinný]
- **`<free>` Volný element pro libovolný obsah**, právě tímto elementem je možné vložit libovolnou datovou strukturu rozšiřující rámec XML Schema pro applety.
[umístění: `//free`, datový typ: rodičovský element pro libovolný obsah, nepovinný]

Element `<visual>` (`//model/visual`)

- **`<filename>` Cesta k ilustračnímu obrázku appletu / modelu**
[umístění: `//model/visual/filename`, datový typ: `xs:string`, povinný]

Element <settings> (//model/settings)

- Množina elementů <setting> **Specifikuje jednu položku nastavení modelu**
[umístění: //model/settings/setting, datový typ: rodičovský element, nepovinný]

Element <setting> (//model/settings/setting)

- <name> **Název položky nastavení**
[umístění: //model/settings/setting/key, datový typ: string-long, povinný]
- <key> **Unikátní** (v rámci položek nastavení modelu / appletu) **identifikátor**
[umístění: //model/settings/setting/key, datový typ: name-32, povinný]
- <description> **Popisek**, měl by obsahovat popis významu položky.
[umístění: //model/settings/setting/description, datový typ: xs:string (doporučeno použít HTML formátování), povinný]
- <default> **Výchozí hodnota**, měla by odpovídat zvolenému datovému typu (viz //model/settings/setting/type)
[umístění: //model/settings/setting/default, datový typ: xs:string, povinný]
- <type> **Požadovaný typ datové hodnoty**, vyplněná hodnota v Administraci bude validována podle příslušného typu.
[umístění: //model/settings/setting/type, datový typ: enumeration {string, integer, double}, výchozí string, povinný]

Element <inputs> (//model/inputs)

- Množina elementů <input> **Specifikuje jeden vstupní bod**
[umístění: //model/inputs/input, datový typ: rodičovský element, nepovinný]

Element <input> (//model/inputs/input)

- <name> **Název položky nastavení**
[umístění: //model/inputs/input/name, datový typ: string-long, povinný]
- <key> **Unikátní** (v rámci bodů modelu / appletu) **identifikátor**
[umístění: //model/inputs/input/key, datový typ: name-32, povinný]
- <require> Určuje, zda **je nutné bod povinně využít** (vést jím spojení) či nikoliv.
[umístění: //model/inputs/input/require, datový typ: xs:boolean, povinný]
- <description> **Popisek**, měl by obsahovat popis vstupního bodu. Tento element není v Administraci použit.

[umístění: //model/inputs/input/description, datový typ: xs:string (doporučeno použít HTML formátování), povinný]

- **<entities>** Obsahuje **seznam entit**, na které bude model správně reagovat
[umístění: //model/inputs/input/entities, datový typ: rodičovský element, povinný]
- **<visual>** Specifikuje **zobrazení bodu v Designeru**
[umístění: //model/inputs/input/visual, datový typ: rodičovský element, povinný]

Element <entities> (//model/inputs/input/entities)

- Množina elementů **<entity>** **Specifikuje jednu entitu**, kterou model přijme
[umístění: //model/inputs/input/entities/entity, datový typ: rodičovský element, nepovinný]

Element <entity> (//model/inputs/input/entities/entity)

- **<key>** **Jednoznační identifikátor entity**
[umístění: //model/inputs/input/entities/entity/key, datový typ: name-32, povinný]

Element <visual> (//model/inputs/input/visual)

- **<location>** **Specifikuje pozici bodu v rámci ilustračního obrázku v Designeru**
[umístění: //model/inputs/input/visual/location, datový typ: rodičovský element, povinný]

Element <location> (//model/inputs/input/visual/location)

- **<x>** **Souřadnice středu bodu v pixelech v rámci ilustračního obrázku v Designeru**
[umístění: //model/inputs/input/visual/location/x, datový typ: xs:nonNegativeInteger, povinný]
- **<y>** **Souřadnice středu bodu v pixelech v rámci ilustračního obrázku v Designeru**
[umístění: //model/inputs/input/visual/location/y, datový typ: xs:nonNegativeInteger, povinný]

Element <outputs> (//model/outputs)

- Množina elementů **<output>** **Specifikuje jeden výstupné bod**
[umístění: //model/outputs/output, datový typ: rodičovský element, nepovinný]

Element <output> (//model/outputs/output)

- <name> **Název položky nastavení**
[umístění: //model/outputs/output/name, datový typ: string-long, povinný]
- <key> **Unikátní (v rámci bodů modelu / appletu) identifikátor**
[umístění: //model/outputs/output/key, datový typ: name-32, povinný]
- <description> **Popisek**, měl by obsahovat popis vstupního bodu. Tento element není v Administraci použit.
[umístění: //model/outputs/output/description, datový typ: xs:string (doporučeno použít HTML formátování), povinný]
- <entities> Obsahuje **seznam entit**, na které bude model správně reagovat
[umístění: //model/outputs/output/entities, datový typ: rodičovský element, povinný]
- <visual> Specifikuje **zobrazení bodu v Designeru**
[umístění: //model/outputs/output/visual, datový typ: rodičovský element, povinný]

Element <entities> (//model/outputs/output/entities)

- Množina elementů <entity> **Specifikuje jednu entitu**, kterou model přijme
[umístění: //model/outputs/output/entities/entity, datový typ: rodičovský element, nepovinný]

Element <entity> (//model/outputs/output/entities/entity)

- <key> **Jednoznační identifikátor entity**
[umístění: //model/outputs/output/entities/entity/key, datový typ: name-32, povinný]

Element <visual> (//model/outputs/output/visual)

- <location> **Specifikuje pozici bodu v rámci ilustračního obrázku v Designeru**
[umístění: //model/outputs/output/visual/location, datový typ: rodičovský element, povinný]

Element <location> (//model/outputs/output/visual/location)

- <x> **Souřadnice středu bodu v pixelech v rámci ilustračního obrázku v Designeru**
[umístění: //model/outputs/output/visual/location/x, datový typ: xs:nonNegativeInteger, povinný]

- **<y> Souřadnice středu bodu v pixelech v rámci ilustračního obrázku v Designeru**
[umístění: //model/outputs/output/visual/location/y, datový typ:
xs:nonNegativeInteger, povinný]

Datový typ „name-32“

Používá se pro jednoznačné identifikátory. Datový typ založený na typu xs:string, s délkou 1 až 32 znaků včetně, povolené pouze alfanumerické ASCII znaky a tečka, začínající písmenem.

Datový typ „string-long“

Používá se pro názvy prvků. Datový typ založený na typu xs:string, s délkou 1 až 256 znaků včetně.

Stromové znázornění:

- model
 - name
 - description
 - visual
 - filename
 - settings
 - setting
 - name
 - key
 - description
 - default
 - type
 - inputs
 - input
 - name
 - key
 - require
 - description
 - entities
 - entity
 - key
 - visual
 - location
 - x
 - y
 - outputs
 - output
 - name
 - key
 - description
 - entities
 - entity
 - key
 - visual
 - location
 - x
 - y
 - free

4 KONFIGURAČNÍ SOUBOR XML SIMULACE

XML Schema je k dispozici v příloze B. Ve schéma jsou základní informace uvedené v části `<xs:annotation>`, v této kapitole budou rozebrány podrobněji. Rozšířené datové typy a strukturované elementy jsou podrobně rozepsány.

Root element `<simulation>` (`//simulation`)

Atribut:

- **@version Verze** generátoru XML, formát verze: 8 čísel + volitelně 1 písmenka
[umístění `//simulation[@version]`, datový typ: `xs:string`, povinný]

Obsahuje podelementy:

- **<name> Název simulace**
[umístění: `//simulation/name`, datový typ: `string-long`, povinný]
- **<saved> Datum a čas uložení konfiguračního souboru** administrací
[umístění: `//simulation/saved`, datový typ: `xs:dateTime`, povinný]
- **<start-date> Datum a čas zahájení simulace**
[umístění: `//simulation/start-date`, datový typ: `xs:dateTime`, povinný]
- **<random-seed> Násada pro generátor náhodných čísel**
[umístění: `//simulation/random-seed`, datový typ: `xs:long`, povinný]
- **<models> Seznam modelů** v simulaci
[umístění: `//simulation/models`, datový typ: rodičovský element, povinný]
- **<connections> Seznam spojení modelů** v simulaci
[umístění: `//simulation/connections`, datový typ: rodičovský element, povinný]
- **<free> Volný element pro libovolný obsah**, právě tímto elementem je možné vložit libovolnou datovou strukturu rozšiřující rámec XML Schema pro simulaci.
[umístění: `//simulation/free`, datový typ: rodičovský element pro libovolný obsah, nepovinný]

Element `<models>` (`//simulation/models`)

- Množina elementů `<model>` **Specifikuje jeden model**, výskyt elementu není povinný, ale simulace bez modelů nemá smysl
[umístění: `//simulation/models/model`, datový typ: rodičovský element, nepovinný]

Element <model> (//simulation/models/model)

Atributy:

- **@unique Unikátní textový identifikátor modelu**
[umístění //simulation/models/model[@unique], datový typ: xs:string, povinný]
- **@unique-designer Unikátní textový identifikátor objektů v Designeru**, obvykle odpovídá //simulation/models/model[@unique]
[umístění //simulation/models/model[@unique-designer], datový typ: xs:string, povinný]

Obsahuje podelementy:

- **<id> ID appletu**, podle kterého je model vytvořen, dle uložení v Administraci
[umístění: //simulation/models/model/id, datový typ: xs:nonNegativeInteger, povinný]
- **<workstation> ID pracovní stanice** v Administraci
[umístění: //simulation/models/model/workstation, datový typ: xs:nonNegativeInteger, povinný]
- **<xmlfile> URL adresa konfiguračního XML soubor pro applet**, podle kterého je model vytvořen
[umístění: //simulation/models/model/xmlfile, datový typ: xs:string, povinný]
- **<name> Název modelu**
[umístění: //simulation/models/model/name, datový typ: string-long, povinný]
- **<ports> Informace o síťových portech**
[umístění: //simulation/models/model/ports, datový typ: rodičovský element, povinný]
- **<settings> Obsahuje jednotlivá nastavení**
[umístění: //simulation/models/model/settings, datový typ: rodičovský element, povinný]

Element <ports> (//simulation/models/model/ports)

- **<server> Specifikuje port**, na kterém naslouchá server
[umístění: //simulation/models/model/ports/server, datový typ: port-type, povinný]
- **<client> Specifikuje port**, na kterém naslouchá model
[umístění: //simulation/models/model/ports/client, datový typ: port-type, povinný]

Element <settings> (//simulation/models/model/settings)

- Množina elementů <setting> **Specifikuje jednu položku nastavení**
[umístění: (//simulation/models/model/settings/setting, datový typ: rodičovský element, nepovinný]

Element <setting> (//simulation/models/model/settings/setting)

- <key> **Klíč, který identifikuje položku**, tato položka se přímo vztahuje k //model/settings/setting/key z kapitoly 3.
[umístění: //simulation/models/model/settings/setting/key, datový typ: name-32, povinný]
- <value> **Hodnota položky**, položka je vždy uložena ve formě řetězce, na jakýkoliv jiný typ je nutné přeparsovat
[umístění: //simulation/models/model/settings/setting/value, datový typ: xs:string, povinný]

Element <connections> (//simulation/connections)

- Množina elementů <connection> **Specifikuje jedno spojení modelů**, výskyt elementu není povinný, ale simulace osamocených modelů nevyžaduje použití distribuované simulace
[umístění: //simulation/connections/connection, datový typ: rodičovský element, nepovinný]

Element <connection> (//simulation/connections/connection)

- <name> **Název spojení**
[umístění: //simulation/connections/connection/name, datový typ: string-long, povinný]
- <from> **Specifikace jednoho konce spojení**
[umístění: //simulation/connections/connection/from, datový typ: xs:nonNegativeInteger, povinný]
- <to> **Specifikace druhého konce spojení**
[umístění: //simulation/connections/connection/to, datový typ: rodičovský element, povinný]

Element <from> (//simulation/connections/connection/from)

- **<unique> Název modelu, ze kterého vede spojení**, viz
//simulation/models/model[@unique]
[umístění: //simulation/connections/connection/from/unique, datový typ: string-32,
povinný]
- **<key> Klíč bodu, ze kterého vede spojení**, viz kapitola 3 –
//model/outputs/output/key
[umístění: //simulation/connections/connection/from/key, datový typ: name-32,
povinný]

Element <to> (//simulation/connections/connection/to)

- **<unique> Název modelu, do kterého vede spojení**, viz
//simulation/models/model[@unique]
[umístění: //simulation/connections/connection/to/unique, datový typ: string-32,
povinný]
- **<key> Klíč bodu, do kterého vede spojení**, viz kapitola 3 – //model/inputs/input/key
[umístění: //simulation/connections/connection/to/key, datový typ: name-32, povinný]

Datový typ „port-type“

Používá se pro číslo portu síťové komunikace. Datový typ založený na typu xs:unsignedShort, s hodnotami od 1024 do 65535 včetně.

Datové typy „name-32“ a „string-long“ jsou shodné s těmi uvedenými k kapitole 3.

Stromové znázornění:

- simulation [@version]
 - name
 - saved
 - start-date
 - random-seed
 - models [@unique, @unique-designer]
 - model
 - id
 - workstation
 - xmlfile
 - name
 - ports
 - server
 - client
 - settings
 - setting
 - key
 - value
 - connection
 - name
 - from
 - unique
 - key
 - to
 - unique
 - key
 - free

5 PŘEDÁVANÉ PARAMETRY APPLETU

Applet je z Administrace načítán pomocí tagu `<applet>`, který je sice zastaralý, avšak obecně a hlavně bezproblémově kompatibilní se všemi internetovými prohlížeči většinově zastoupenými na trhu. Ukázka kódu (zalomení textu v URL adresách je pouze z důvodu formátování textu):

```
<applet      codebase="http://localhost/www/../../SimulaceBase/build/classes/"
            code="run/AppletDalnice.class"
            width="800"
            height="480"
>
    <param name="wmode" value="opaque" />
    <param name="cache_option" value="no">
    <param name="unique" value="9b3ad4536215377f9fc3e923d6e461e8" />
    <param name="start.date" value="2013-03-11 20:00:00" />
    <param name="random.seed" value="1367505397000" />
    <param name="config.simulation"
            value="http://localhost/
            www/xml/simulations/simulation-00000043.xml?_=1367522219"
    />
    <param name="config.applet"
            value="http://localhost/
            www/xml/dalnice.stred.xml?_=1367522219"
    />
</applet>
```

- **wmode="opaque"**, tento parametr je pouze pro samotný prohlížeč. Určité verze prohlížečů podle něho vykreslují obsah vnořených prvků / appletů ve stránce jako `<applet>`, `<object>`, aj. Teoreticky jím lze zakázat výchozí stav, kdy vnořený prvek je vždy v popředí.
- **cache_option="no"**, parametr zakáže interní cache Javy³.
- **unique**, parametr identifikující konkrétní načítaný model, podle tohoto parametru naleznete odpovídající záznam v XML konfiguračním souboru simulace, viz 3 – `//simulation/models/model[@unique]`.
- **start.date**, datum a čas zahájení simulace, formát YYYY-MM-DD HH:MM:SS.
- **random.seed**, násada generátoru náhodného čísla, typ *long*⁴, hodnota 0 vyžaduje vygenerovat zcela unikátní hodnoty.
- **config.simulation**, unikátní URL konfiguračního XML souboru konfigurace simulace.
- **config.applet**, unikátní URL konfiguračního XML souboru dotyčného appletu. Tato hodnota je taktéž dostupná z **config.simulation** v umístění

³ Testováno, chová se „svěhlavě“, z tohoto důvodu jsou načítané URL adresy vždy originální.

⁴ Obecně datové 64bitové číslo zapsané ve formě řetězce v dekadické soustavě.

//simulation/models/model[@unique=**unique**]/xmlfile ... tato hodnota je „statická“, tj. adresa není originální na rozdíl od **config.applet**.

6 KOMUNIKACE ADMINISTRACE SE SERVEREM

Dodávaný server komunikuje níže uvedeným způsobem.

Zjištění stavu server

`http://localhost/www/ext/server.send.php?req=service&type=state`

Návratová hodnota XML (ukázka):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<server>
  <timestamp type="date">
    <unix>1367426195296</unix>
    <string>Wed May 01 18:36:35 CEST 2013</string>
  </timestamp>
  <service type="state">
    <state>running</state>
    <log>
      <maxLevelAccepted>9</maxLevelAccepted>
    </log>
  </service>
</server>
```

Všechny ostatní stavy `//server/service/state` jsou pokládány za chybné. Stejně je pokládáno za chybu, pokud se server nepodaří kontaktovat.

Zastavení serveru

`http://localhost/www/ext/server.send.php?req=stop&con=udp`

Neočekává se odpověď, stav se zjistí výše uvedeným způsobem.

Spuštění serveru

`http://localhost/www/ext/server.start.php`

Neočekává se odpověď, stav se zjistí výše uvedeným způsobem.

Zjištění klientů

`http://localhost/www/ext/server.send.php?req=service&type=clients`

Návratová hodnota XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<server>
  <timestamp type="date">
    <unix>1367427024292</unix>
    <string>Wed May 01 18:50:24 CEST 2013</string>
  </timestamp>
  <servise type="clients">
    <client>
      <name>Občerstvení pravé 2176:60031</name>
      <port>60031</port>
      <statistics>
        <messages>
          <input>3</input>
          <inputDateLast type="date">
            <unix>1367427022306</unix>
            <string>Wed May 01 18:50:22 CEST 2013</string>
          </inputDateLast>
          <output>3</output>
          <outputDateLast type="date">
            <unix>1367427019588</unix>
            <string>Wed May 01 18:50:19 CEST 2013</string>
          </outputDateLast>
        </messages>
      </statistics>
    </client>
    <client>
      <name>Dálnice středový prvek 7540:60029</name>
      <port>60029</port>
      <statistics>
        <messages>
          <input>8</input>
          <inputDateLast type="date">
            <unix>1367427023616</unix>
            <string>Wed May 01 18:50:23 CEST 2013</string>
          </inputDateLast>
          <output>9</output>
          <outputDateLast type="date">
            <unix>1367427023908</unix>
            <string>Wed May 01 18:50:23 CEST 2013</string>
          </outputDateLast>
        </messages>
      </statistics>
    </client>
  </servise>
</server>
```

7 NÁZVY MODELŮ

Název model je složen ze samotného názvu modelu (dle kapitoly 4 – //simulation/models/model/name) a portu, na kterém komunikuje se serverem (dle kapitoly 4 – //simulation/models/model/ports/client). Spojovací prvek je dvojtečka „:“. Tímto způsobem je možné zjistit z konfiguračního souboru simulace názvy všech modelů v rámci distribuované simulace.

Příklad názvu modelu s názvem „Model A“ a přiřazeném portu „60023“:

```
Model A:60023
```

Výše uvedený řetězec, tak jak výše uveden (13 znaků) je jednoznačný identifikující název klienta v rámci serveru. Server je *case-sensitive*.

PŘÍLOHA A – XML SCHEMA APLETU

```
<!-- root//xml/schemas/ModelSchema.xsd -->

<?xml version="1.0"?>

<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xs:element name="model">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string" />
        <xs:element name="description" type="xs:string" />
        <xs:element name="visual">
          <xs:annotation>
            <xs:documentation>Pro vizualizaci modelu /
návrh</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="filename" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="settings">
          <xs:annotation>
            <xs:documentation>Možnosti nastavení modelu</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="setting" minOccurs="0"
maxOccurs="unbounded">
                <xs:annotation>
                  <xs:documentation>Položka v nastavení</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name" type="string-long" />
                    <xs:element name="key" type="name-32" />
                    <xs:element name="description" type="xs:string" />
                    <xs:element name="default" type="xs:string" />
                    <xs:element name="type" default="string">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:enumeration value="string" />
                          <xs:enumeration value="integer" />
                          <xs:enumeration value="double" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="inputs">
    <xs:complexType>
      <xs:sequence>
```

```

        <xs:element name="input" type="input-type" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="outputs" >
    <xs:complexType>
        <xs:sequence>
            <xs:element name="output" type="output-type" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="free" minOccurs="0" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:any processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:simpleType name="string-long">
    <xs:annotation>
        <xs:documentation>Název musí být min 1 znaků dlouhý, tj. musí být
vyplněn</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:minLength value="1" />
        <xs:maxLength value="256" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="name-32">
    <xs:annotation>
        <xs:documentation>Název může být 1-32 znaků, tj. musí být
vyplněn</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:minLength value="1" />
        <xs:maxLength value="32" />
        <xs:pattern value="[a-zA-Z]+[a-zA-Z0-9\.]*" />
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="input-type">
    <xs:annotation>
        <xs:documentation>Informace o přichozích
entitách/spojit</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="name" type="string-long" />
        <xs:element name="key" type="name-32" />
        <xs:element name="require" type="xs:boolean" />
        <xs:element name="description" type="xs:string" />
        <xs:element name="entities" type="entities-type"
maxOccurs="unbounded" />
        <xs:element name="visual" type="visual-io-type" />

```

```

    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="output-type">
    <xs:annotation>
      <xs:documentation>Informace o odchozích
entitách/spojít</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="name" type="string-long" />
      <xs:element name="key" type="name-32" />
      <xs:element name="description" type="xs:string" />
      <xs:element name="entities" type="entities-type"
maxOccurs="unbounded" />
      <xs:element name="visual" type="visual-io-type" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="entities-type">
    <xs:annotation>
      <xs:documentation>Toto je informace o entitě</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="entity" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:annotation>
              <xs:documentation>Defakto název elementu</xs:documentation>
            </xs:annotation>
            <xs:element name="key" type="name-32" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="visual-io-type">
    <xs:annotation>
      <xs:documentation>Pro vizualizaci modelu / návrh</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="location">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="x" type="xs:nonNegativeInteger" />
            <xs:element name="y" type="xs:nonNegativeInteger" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

</xs:schema>

```

PŘÍLOHA B – XML SCHEMA SIMULACE

```

<!-- root//xml/schemas/SimulationSchema.xsd -->
<?xml version="1.0"?>

```

```

<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xs:element name="simulation">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="string-long" />
        <xs:element name="saved" type="xs:dateTime" />
        <xs:element name="start-date" type="xs:dateTime" />
        <xs:element name="random-seed" type="xs:long" />
        <xs:element name="models">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="model" type="model-type" minOccurs="0"
maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="connections" type="connections-type" />
        <xs:element name="free" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:any processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>

      <xs:attribute name="version">
        <xs:simpleType>
          <xs:annotation>
            <xs:documentation>Formát verze: 8 čísel + volitelně 1
písmenka, tedy: 20030402b ... první verze</xs:documentation>
          </xs:annotation>
          <xs:restriction base="xs:string">
            <xs:minLength value="8" />
            <xs:maxLength value="9" />
            <xs:pattern value="[0-9]{8}[a-zA-Z]?" />
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>

    </xs:complexType>
  </xs:element>

  <xs:complexType name="model-type">
    <xs:annotation>
      <xs:documentation>Specifikace nastavení 1 modelu</xs:documentation>
    </xs:annotation>

    <xs:sequence>
      <xs:element name="id" type="xs:nonNegativeInteger">
        <xs:annotation>
          <xs:documentation>Zásadní pro zparsování s modelovými xml
soubory, načítají se z DB</xs:documentation>
        </xs:annotation>

```



```

    </xs:element>
    <xs:element name="workstation" type="xs:nonNegativeInteger">
      <xs:annotation>
        <xs:documentation>Id přiřazené pracovní
stanice</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="xmlfile" type="xs:string">
      <xs:annotation>
        <xs:documentation>Pouze pomocná hodnota vhodná pro vizuální
inspekci</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="name">
      <xs:annotation>
        <xs:documentation>Název modelu</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="1" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="ports">
      <xs:annotation>
        <xs:documentation>Nastavení portů</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="server" type="port-type">
            <xs:annotation>
              <xs:documentation>Port pro serverový prvek musí být (jiná
možnost v Javě není) pro jednu simulaci stejný</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="client" type="port-type" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="settings" type="settings-type" />
  </xs:sequence>

  <xs:attribute name="unique" type="string-32" use="required" />
  <xs:attribute name="unique-designer" type="string-32" use="required" />
</xs:complexType>

<!-- ===== -->

<xs:complexType name="settings-type">
  <xs:sequence>
    <xs:element name="setting" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Položka v nastavení, pouze data pro
model</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="key" type="name-32" />
          <xs:element name="value" type="xs:string" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="connections-type">
  <xs:sequence>
    <xs:element name="connection" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>Spojení klient od -> klient do /
MODEL[UNIQUE]</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="string-long" />
          <xs:element name="from" type="connection-point" />
          <xs:element name="to" type="connection-point" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="connection-point">
  <xs:sequence>
    <xs:annotation>
      <xs:documentation>Key označuje konkrétní bod podle
ModelSchema.xsd</xs:documentation>
    </xs:annotation>
    <xs:element name="unique" type="string-32" />
    <xs:element name="key" type="name-32" />
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="port-type">
  <xs:restriction base="xs:unsignedShort">
    <xs:minInclusive value="1024" />
    <xs:maxExclusive value="65535" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="string-long">
  <xs:annotation>
    <xs:documentation>Název musí být min 1 znaků dlouhý, tj. musí být
vyplněn</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="256" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="name-32">
  <xs:annotation>
    <xs:documentation>Název může být 1-32 znaků, tj. musí být
vyplněn</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="1" />
    <xs:maxLength value="32" />
    <xs:pattern value="[a-zA-Z]+[a-zA-Z0-9\.\.]*" />
  </xs:restriction>
</xs:simpleType>

```

```
</xs:simpleType>

<xs:simpleType name="string-32">
  <xs:annotation>
    <xs:documentation>Unique 32 znaků</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:minLength value="32" />
    <xs:maxLength value="32" />
    <xs:pattern value="[a-zA-Z0-9]{32}" />
  </xs:restriction>
</xs:simpleType>

</xs:schema>
```