

Ref : Représentation XML Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08 Janvier 2023 Version : 0A Service : Polytech Marseille Etat : Préliminaire
--	-------------------------------	--

Editeur d'automates: “AutomatesLab”

Annexe du dossier de spécification: Représentation XML des Automates

Référence : SPC-21011201-0B
Fournisseur
Date : 10 Janvier 2023
Version/Édition : 0A
État : Préliminaire

Type de diffusion : Diffusion publique
Autre référence :

Ref : Représentation XML Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08 Janvier 2023 Version : 0A Service : Polytech Marseille Etat : Préliminaire
--	-------------------------------	--

Table des matières

Introduction	3
1. Spécifications de la représentation XML	3
1.1. Sur le format XML	3
1.2. Représentation	3
1.2.1. Description générale	3
1.2.2. Description des balises	4
1.2.2.1. AutomateFile	4
1.2.2.2. Automate	4
1.2.2.3. State	4
1.2.2.4. Transition	5
2. Exemple	6
3. Template XML d'Automate vide	7

Ref : Représentation XML Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08 Janvier 2023 Version : 0A Service : Polytech Marseille Etat : Préliminaire
--	-------------------------------	--

Introduction

Ce document a pour but de spécifier le format XML qui représente les automates gérés par l'application AutomatesLab. Il contient une description détaillée du format du document, des balises autorisées et les associées, ainsi qu'un exemple concret de représentation.

Ce document décrit la version 1.0 de la spécification.

1. Spécifications de la représentation XML

1.1. Sur le format XML

Les fichiers XML représentant un automate suivrons bien entendu les normes dudit langage de balisage. Pour plus d'informations sur ces normes, consulter le [site officiel](#).

Concernant notre logiciel, la norme plus importante est la suivante:

- Les caractères spécifiques au XML (chevrons ouvrants, fermants, etc) sont remplacés par leur références d'entités correspondantes (par exemple, le chevrons ouvrant '<' est codé par la référence "<")

En effet, les chevrons étant souvent utilisé dans les langages informatiques, il s'agit d'un caractère assez important.

1.2. Représentation

1.2.1. Description générale

Un fichier représentant un automate commence toujours par la balise d'information xml suivante:

```
<?xml version="1.0" encoding="UTF-8"?>
```

L'encodage du fichier est donc UTF-8. L'ordre des attributs dans une balise n'importe pas. Les seules balises acceptées sont les suivantes:

- "AutomateFile" unique, comme racine ;
- "Automate" unique, représente un automate ;
- "State" absente ou multiple, représente un état de l'automate ;
- "Transition" absente ou multiple, représente une transition partant d'un état. Elle est obligatoirement auto-fermante.

Ce sont les seules balises acceptées.

Ref : Représentation XML Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08 Janvier 2023 Version : 0A Service : Polytech Marseille Etat : Préliminaire
--	-------------------------------	--

1.2.2. Description des balises

1.2.2.1. AutomateFile

La racine d'un fichier contenant un Automate est la balise "AutomateFile". Elle a l'attribut "checksum" qui prend la valeur (décimale) de la checksum calculée par le logiciel lorsqu'il enregistre le fichier. Cette checksum est calculée en prenant en entrée le string formé par toutes les sous-balises de la racine. Cela permet de vérifier que le fichier n'a pas été modifié en dehors de l'utilisation normale du logiciel.

Exemple:

```
<AutomateFile checksum="2147483647"></AutomateFile>
```

1.2.2.2. Automate

La balise "Automate" permet d'encapsuler des états. Cette balise n'a pas d'attributs. Elle existe afin de laisser la possibilité d'étendre le logiciel dans de futures mises à jour.

Exemple:

```
<Automate></Automate>
```

1.2.2.3. State

Un automate est constitué d'une collection d'états. Une balise "Automate" contient donc un certain nombre de balises "State". Une balise state peut avoir jusqu'à 3 attributs:

- Un attribut "number" qui correspond à son identifiant. Il est donc unique ; cette balise est obligatoire et ne peut contenir que des chiffres.
- Deux attributs "X" et "Y" qui contiennent la position de l'automate sur l'affichage graphique, en pixels (deux entiers naturels). Ils sont facultatifs, et s'ils sont absents, valent "0" par défaut.
- Un attribut "isInitial" qui indique si l'état est initial ou non. Il ne peut valoir que "true" ou "false". Il n'est pas obligatoire, et s'il est absent, la valeur par défaut est "false".
- Un attribut "isFinal" qui indique si l'état est final ou non. Il ne peut valoir que "true" ou "false". Il n'est pas obligatoire, et s'il est absent, la valeur par défaut est "false".

Exemples:

- ```
<State number="0" isInitial="true" X="15" Y="12">
```
- ```
<State number="0" isInitial="true" isFinal="false"></State>
```
- ```
<State number="15" Y="12">
```
- ```
<State number="1" isFinal="true">
```

Ref : Représentation XML Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08 Janvier 2023 Version : 0A Service : Polytech Marseille Etat : Préliminaire
--	-------------------------------	--

1.2.2.4. Transition

Un état peut posséder plusieurs transitions. Une balise “State” peut alors contenir un certain nombre de balises “Transition” qui partent de cet état. Celles-ci ne contiennent pas plus d’objets et sont donc auto-fermantes. Une transition a les attributs suivants:

- Une “destination”, qui représente l’état d’arrivée de la transition. Cet attribut a donc les mêmes propriétés que le “number” des balises “State” qui ne peut être qu’un entier. Cet attribut est obligatoire et ne peut pas valoir le String vide.
- Des lettres acceptées par transition “letters”. Elles sont représentées par un String (chaîne de caractères), dont chaque caractère représente une lettre de transition différente. Cet attribut est facultatif, et s’il est absent, vaut la chaîne de caractères vide.
- Un attribut “acceptsEmptyWord”, qui est un booléen qui signifie si la transition accepte ou non le mot vide. C’est un booléen peut valoir “true” ou “false”, mais la chaîne de caractères vide est aussi acceptée. Cet attribut est facultatif, et est évalué à “false” s’il est absent ou vaut la chaîne de caractères vide.

Exemples:

- `<Transition destination="0" letters="b"/>`
- `<Transition destination="1" letters="abde" acceptsEmptyWord="true"/>`
- `<Transition destination="0"/>`

Ref : Représentation XML Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08 Janvier 2023 Version : 0A Service : Polytech Marseille Etat : Préliminaire
--	-------------------------------	--

2. Exemple

On donne l'exemple de l'automate ci-dessous:

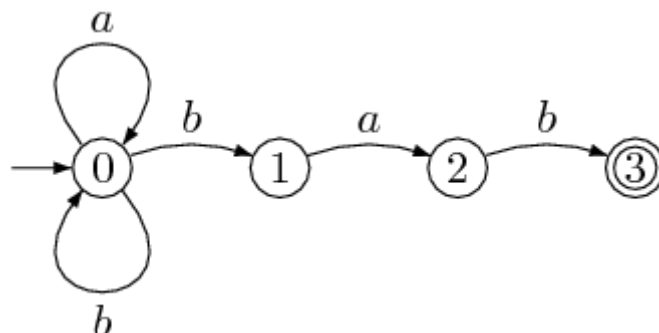


Fig. 1: Automate fini

Selon la spécification décrite plus haut, cet automate sera traduit par le code XML suivant:

```

<?xml version="1.0" encoding="UTF-8"?>
<AutomateFile checksum="57">

  <Automate>
    <State number="0" isInitial="true" X="15" Y="25">
      <Transition destination="0" letters="ab"/>
      <Transition destination="1" letters="b" acceptsEmptyWord="false"/>
    </State>
    <State number="1" X="35" Y="25">
      <Transition destination="2" letters="a"/>
    </State>
    <State number="2" X="65" Y="25">
      <Transition destination="3" letters="b"/>
    </State>
    <State number="3" isFinal="true" X="95" Y="25">
    </State>
  </Automate>

</AutomateFile>

```

Ref : Représentation XML Emetteur : Yves Jehanno Client : Nicolas Baudru Projet : Editeur d'automates	Projet Editeur d'automates	Date: 08 Janvier 2023 Version : 0A Service : Polytech Marseille Etat : Préliminaire
--	-------------------------------	--

3. Template XML d'Automate vide

Dans sa version actuelle, le template chargé par défaut par le logiciel est le suivant:

```
<?xml version="1.0" encoding="UTF-8"?>
<AutomateFile checksum="7691456007008892058">
  <Automate>
  </Automate>
</AutomateFile>
```