Introduction
Objectifs
Application
Contraintes
Stratégie
Prototypes
Tests
Conclusion
Langage
Recomms
Evaluation
Contacts

LaFoSec: Étude de la sécurité intrinsèque des langages fonctionnels [1]

Partie IV sur IV

Développement d'un valideur XML en OCaml

Damien Doligez, Christèle Faure, Thérèse Hardin, Manuel Maarek

JFLA - février 2013

SAFE RIVER
Safety & Security Forge

1. Etude commanditée par l'Agence Nationale de la Sécurité des Systèmes d'Information

Développement d'une application de sécurité en OCaml

Objectifs

- Pertinence de l'utilisation du langage OCaml pour le développement d'applications de sécurité
- Pertinence et applicabilité des recommandations de sécurité issues de l'étude du langage OCaml
- Faisabilité d'une évaluation de sécurité d'un logiciel écrit en OCaml

Application

XSVGen

Générateur de valideurs XML vis-à-vis de grammaires XSD (XML Schema)

- Conformément aux recommandations XML du W3C
- Conformément aux recommandations XSD du W3C
- Interprétation restrictive des langages XML et XSD pour souscrire à des objectifs de sécurité

La sécurité des langages XML/XSD est hors du sujet de l'étude

Cadre d'utilisation

Protection de services prenant en entrée des fichiers XML

Exemples XSD

Exemple de définition d'un type simple XSD

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="exemple-type-simple">
     <xsd:element name="test" type="Test">
     <xsd:simpleType name="Test">
          <xsd:union>
               <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                         <xsd:maxLength value="3">
                         <xsd:pattern value="a+">
               <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                         <xsd:maxLength value="3">
                         <xsd:pattern value="b+">
```

Exemple de définition d'un type complexe XSD

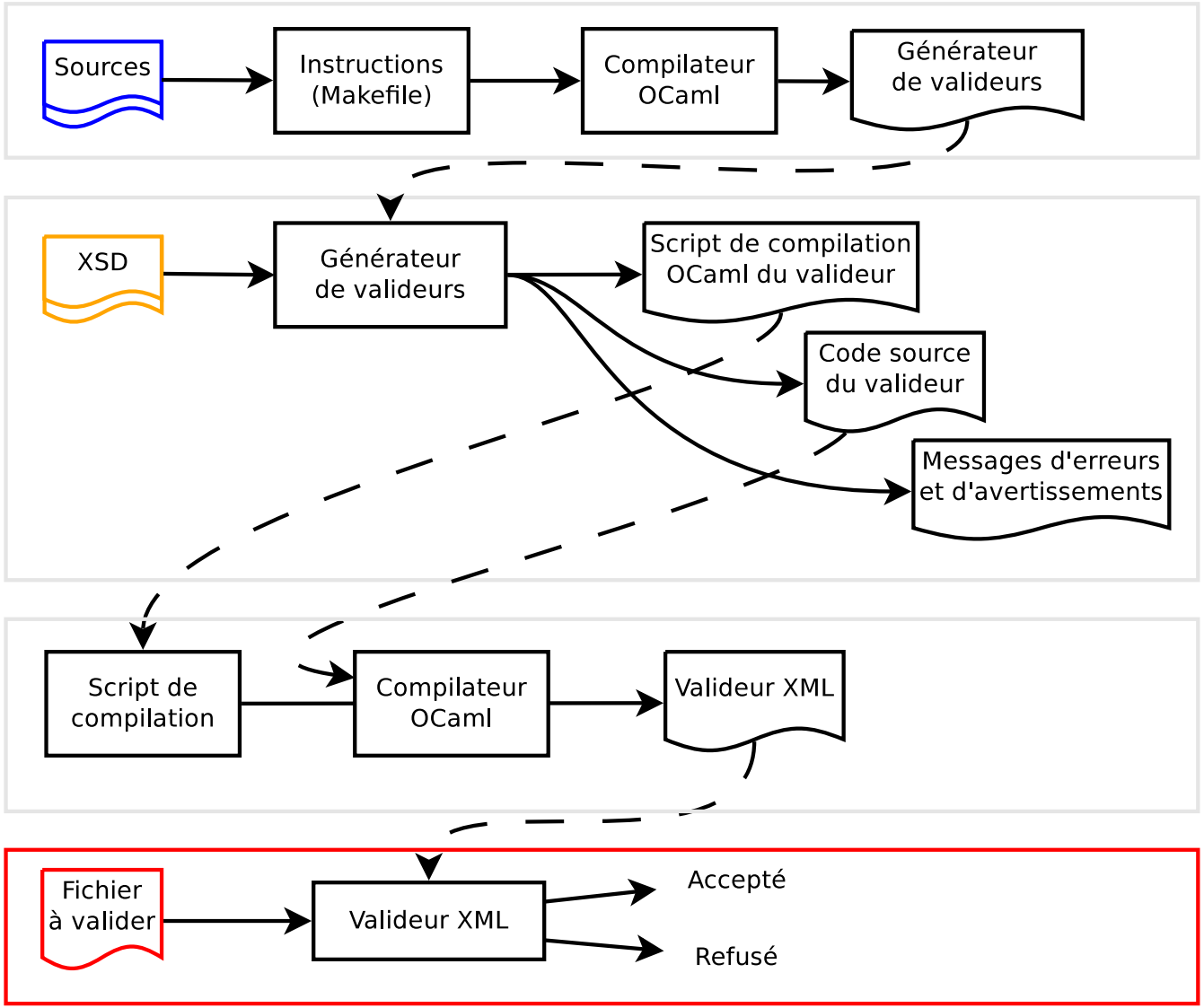
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="exemple-type-complexe">
        <xsd:complexType name="base">
             <xsd:all>
                  <xsd:element name="e1">
                  <xsd:element name="e2">
        <xsd:element name="doc">
             <xsd:complexType>
                  <xsd:complexContent>
                       <xsd:extension base="base">
```

Contraintes

Contraintes du développement

- Conformité aux recommandations émises par l'étude du langage OCaml
- Préparation de l'évaluation : la recherche de vulnérabilités
  - documents de spécification, conception et développement
  - base de tests en robustesse, en conformité

Fonctionnement de l'application



Développement incrémental
Prototypes

Prototypes	Fonctionnalités			
	Caractères	Types de données	Types de structures	Méta constructions
Prototype 1	UTF-8	expressions rationnelles	combinateurs (*all choice sequence*)	
Prototype 2	entités	types primitifs, facettes	types complexes (*extension*, *restriction*)	*attributeGroup*
Prototype 3		types construits		importation, groupes concrets, héritage
Prototype final v0.2	Prototype fonctionnellement complet			

Prototype final v0.2

Fonctionnalités

XML	supports	versions 1.0 et 1.1 — encodage UTF-8 — entités
	restrictions	caractères déconseillés par XML 1.1 — type de document — instructions de traitement — sections d'échappement — commentaires — références non-ASCII — espaces de noms
	limitations	équivalences Unicode — clés XML
XSD	supports	version 1.1 — types simples prédéfinis (primitifs, construits) — domaines de valeurs — expressions rationnelles — types complexes — combinateurs — groupes — importations — occurrences — espaces de noms
	restrictions	entrelacement — attributs d'instance — URI de fichiers XSD
	limitations	vérifications de validité XSD (contraintes d'héritage, validité des valeurs par défaut) — groupes d'attributs par défaut — modularité XSD — certaines catégories de caractères — expressions de contraintes (chemins, valeurs, précisions) — clés XSD

Résultats de la validation sur les bases de tests

Nom	Nombre	Résultats	Origine
UTF-8	280	100%	Développée pour l'étude
XML	312	100%	W3C XML Test Suite (extrait)
XSD	74000	92% [2] 99% [3]	XML Schema Test Suite (extrait)
XSD-LIM	159	100%	Développée pour l'étude

2. des grammaires XSD
3. des fichiers XML

Apports du langage

Pour l'implémentation des exigences de sécurité

- Invariants de représentation de données : typage, encapsulation
- Protection des données : types non mutables, encapsulation
- Encapsulation des effets de bord

Pour l'implémentation des exigences fonctionnelles

- Cloisonnement des fonctionnalités : modules distincts
- Traitements pas cas : filtrage exhaustif et non fragile
- Application partielle pour isoler les étapes de traitement
- Explicitation du flot d'exécution

Apports des recommandations

Pour l'implémentation des exigences de sécurité

- Spécification des interfaces
- Encapsulation des chaînes de caractères
- Gestion des exceptions
- Encapsulation des effets de bord
- Utilisation de types non mutables
- Filtrage exhaustif et non fragile

Pour la facilité de développement

- Vérification de la complétude des traitements
- Développement incrémental facilité

Résultats de l'évaluation

Difficultés d'évaluation rencontrées

- Manque d'outils d'analyse de OCaml
- Evaluateurs peu formés à OCaml

Conclusions

- Utilisation fragile de `Sys.command` : non spécifique à OCaml
- Valideur reconnu fonctionnellement fiable et robuste
- Détection de fichiers invalides non détectés par d'autres valideurs
- L'inverse n'étant pas vrai

Liste des participants

Tranches conditionnelles du projet LaFoSec : développement et évaluation

- Développement de XSVGen : SafeRiver
  - Samuel Colin
  - Damien Doligez
  - Christèle Faure
  - Thérèse Hardin
  - Manuel Maarek
- *Cible de sécurité et évaluation du valideur : Oppida*

Contacts

- Véronique Delebarre : veronique.delebarre@safe-river.com
- Christèle Faure : christele.faure@safe-river.com