**Mục lục**

[**I.** **Tìm hiểu XML** 1](#_Toc523388059)

[**1.** **Giới thiệu XML** 1](#_Toc523388060)

[**2.** **Sử dụng XML ?** 2](#_Toc523388061)

[**3.** **XML tree** 2](#_Toc523388062)

[**II.** **Tìm hiểu XACML** 3](#_Toc523388063)

[**1.** **Giới thiệu XACML** 3](#_Toc523388064)

[**2.** **Cấu trúc XACML** 4](#_Toc523388065)

[**3.** **Thành phần của XACML** 6](#_Toc523388066)

[**3.1.** **Rule**: 7](#_Toc523388067)

[***3.1.1.*** **Target** 7](#_Toc523388068)

[***3.1.2.*** **Effect** 8](#_Toc523388069)

[***3.1.3.*** **Condition (*Điều kiện* )** 8](#_Toc523388070)

[***3.1.4.*** **Biểu thức *obligation*(nghĩa vụ)** 8](#_Toc523388071)

[***3.1.5.*** **Biểu thức *advice (tư vấn)*** 8](#_Toc523388072)

[**3.2.** **Policy** 8](#_Toc523388073)

[***3.2.1.*** **Mục tiêu chính sách (target policy)** 8](#_Toc523388074)

[***3.2.2.*** **Rule-combining algorithm-identifier ( thuật toán kết hợp quy tắc)** 9](#_Toc523388075)

[***3.2.3.*** **Biểu thức Obligation (nghĩa vụ)** 9](#_Toc523388076)

[***3.2.4.*** **Biểu thức advice (tư vấn)** 9](#_Toc523388077)

[**3.3.** **Policy set (bộ chính sách)** 9](#_Toc523388078)

[***3.3.1.*** **Policy-combining algorithm (Thuật toán kết hợp chính sách)** 9](#_Toc523388079)

[***3.3.2.*** **Biểu thức Obligation (nghĩa vụ)** 9](#_Toc523388080)

[***3.3.3.*** **Biểu thức Advice** 10](#_Toc523388081)

[**III.** **Logic Vị từ** 10](#_Toc523388082)

[**1.** **The Logic of XACML** 10](#_Toc523388083)

[***1.1.*** **Xacml** 10](#_Toc523388084)

[**1.2.** **Three-Valued Lattice (Lưới ba giá trị)** 10](#_Toc523388085)

[**1.3.** **The Match Evaluation** 10](#_Toc523388086)

[**1.4.** **Condition Evaluation** 11](#_Toc523388087)

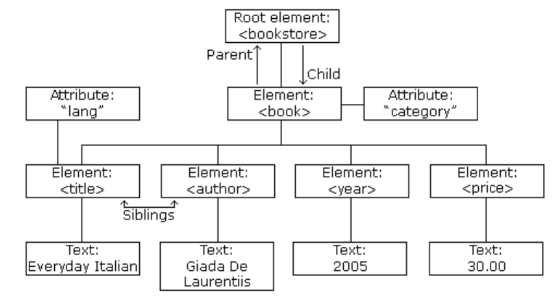
[**1.5.** **Extended Values** 11](#_Toc523388088)

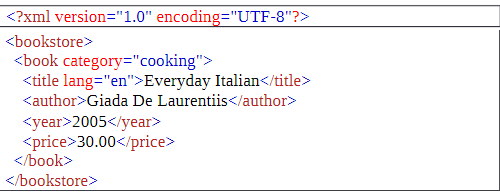
[**1.6.** **Biểu thức rút gọn:** 12](#_Toc523388089)

1. **Tìm hiểu XML**
   1. **Giới thiệu XML**
      1. **XML là gì ?**
      * XML là viết tắt của cụm từ (eXtensible Markup Language).
      * XML được thiết kế để lưu trữ và truyền dữ liệu.
      * XML được thiết kế để con người và máy tính đều có thể đọc được.
      * XML là ngôn ngữ đánh dấu mở rộng (giống như HTML) do W3C đề nghị với mục đích tạo ra các ngôn ngữ đánh dấu khác.
      * Đây là một tập con đơn giản có thể mô tả nhiều loại dữ liệu khác nhau.
      * Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với internet.
      1. **Sự khác nhau giữa XML và HTML ?**
      * XML được thiết kế để mang dữ liệu còn HTML được thiết kế để hiển thị dữ liệu.
      * XML không sử dụng các tag được định nghĩa trước như html
   2. **Sử dụng XML ?**
2. **XML được sử dụng**
   * + XML được sử dụng trong nhiều khía cạnh của phát triển web.
     + XML thường được sử dụng để tách biệt dữ liệu và việc trình bày dữ liệu:

* XML không chứa thông tin về việc làm thế nào để hiện dữ liệu trong XML có được trình bày theo nhiều cách khác nhau. Do đó XML có sự tách biệt hoàn toàn giữa dữ liệu và sự trình bày dữ liệu đó.

1. **XML tách dữ liệu từ HTML**
   * + khi hiện thị dữ liệu dưới dạng HTML, chúng ta không cần phải chỉnh sửa các tập tin HTML khi dữ liệu thay đổi. Với XML, dữ liệu có thể được lưu trữ trong các tập tin XML riêng biệt.
   1. **XML tree**
      * XML được tạo bởi các tag ( XML element) có thể lồng lên nhau, tag ngoài ta gọi là tag cha và các tag cha và các tag bên trong gọi là tag con và chúng có thể được tổ chức theo cấu trúc cây thư mục.



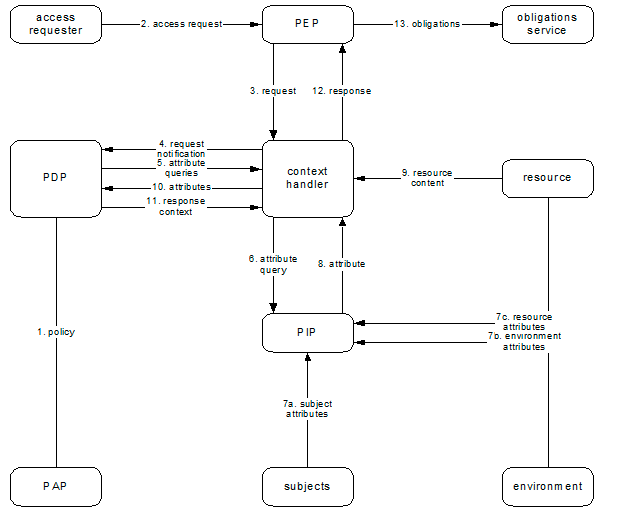
Với sơ đồ trên ta có XML như sau:

1. **Tìm hiểu XACML**
2. **Giới thiệu XACML**

* eXtensible Access Control Markup Language được viết là XACML.
* XACML là một tài liệu từ Oasis được hình thành để củng cố sự nổ lực các bên liên quan như IBM và Trường đại học Milan. Nó được sử dụng để kết hợp với SAML và đem đến một ý nghĩa lớn trong việc tiêu chuẩn hóa các quyết định kiểm soát truy cập cho tài liệu XML. XACML còn được gọi là XACL được dùng để xác định có nên cho phép truy cập đến nguồn tài nguyên hay là toàn bộ một tài liệu, hay trên nhiều tài liệu hay một phần của tài liệu.
* XACML phiên bản 2.0 đã được chấp nhận như một tiêu chuẩn OASIS cùng với sáu cấu hình của XACML: SAML 2.0, XML Digital Signature, Privacy Policy (chính sách bảo mật), Hierarchical Resource (phân cấp tài nguyên) và RBAC (Role-Based Access Control). XACML là một tiêu chuẩn bổ sung của OASIS để đưa ra các quyết định việc điều khiển truy cập
* XACML phiên bản 3.0 có các cấu hình cho các quy định Kiểm soát xuất khẩu (EC), Kiểm soát sở hữu trí tuệ (IP), chuẩn hóa các lược đồ thuộc tính và hơn thế nữa, ví dụ như cho phép các tổ chức có triển khai XACML khác nhau cộng tác dễ dàng hơn. Với XACML 3.0, cấu hình SAML hiện có cũng được nâng cao.

1. **Cấu trúc XACML**

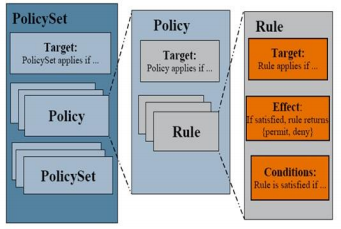
* Mô hình XACML:



* Bảng viết tắt:

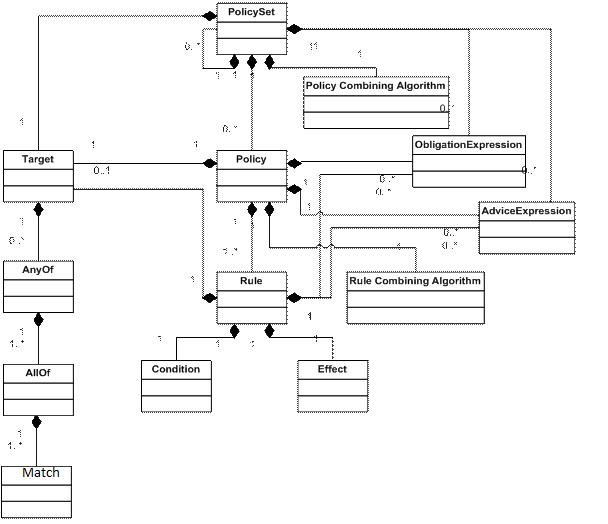
|  |  |  |  |
| --- | --- | --- | --- |
| Tên viết tắt | Tên tiếng anh | Tên tiếng việt | Mô tả |
| PEP | Policy Enforcement Point | Điểm thực thi chính sách | Thực hiện kiểm soát truy cập bằng cách yêu cầu quyết định và thực thi các quyết định ủy quyền |
| PAP | Policy Administration Point | Điểm quản lý chính sách | Tạo và lưu trữ chính sách bảo mật |
| PDP | The Policy Decision Point | Điểm quyết định chính sách | Nhận, xem xét yêu cầu. Sau đó áp dụng các chính sách cùng với việc đánh giá các chính sách đó rồi trả về PEP |
| PIP | Policy Information Point | Điểm thông tin chính sách | Là nguồn gốc của các giá trị thuộc tính hoặc các dữ liệu cần thiết để đánh giá chính sách |
|  | Context Handler | Bối cảnh xử lý | Xác định để chuyển đổi các yêu cầu theo định dạng gốc của nó với hình thức XACML và chuyển đổi các quyết định ủy quyền theo hình thức XACML sang định dạng gốc |

* Mô hình hoạt động theo các bước sau:

1. ***PAPs*** viết  ***policies*** (chính sách) and ***policy sets (bộ chính sách)*** và cung cấp chúng cho ***PDP.*** Các ***policies*** (chính sách) hoặc ***policy sets***  (bộ chính sách )thể hiện ***policy***  hoàn chỉnh cho mộ ***targe (mục tiêu)***được chỉ định.
2. Người yêu cầu ***access***  (***truy cập)*** gửi yêu cầu ***access*** (***truy cập***) vào ***PEP***
3. Các PEP gửi yêu cầu cho access (truy cập) đến context handler (xử lý bối cảnh) ở định dạng yêu cầu có nguồn gốc của nó, tùy chọn bao gồm attributes của **subjects**, **resource**  , ***action***, **environment**  và các thể loại khác.
4. ***Context handler***  (xử lý bối cảnh) xây dựng một  ***context (bối cảnh***) yêu cầu XACML, ùy chọn thêm các thuộc tính và gửi nó đến  ***PDP***
5. Các ***PDP*** yêu cầu bất kỳ thêm **subjects**, **resource**  , ***action***, **environment**  và các thể loại khác (không hiển thị) **attributes**  từ ***context handler***
6. ***Context handler***  yêu cầu các **attributes**  từ một ***PIP***
7. Các ***PIP*** có được các yêu cầu **attributes**
8. Các ***PIP*** trả về yêu cầu ***các* attributes**  cho ***Context handler***
9. Tùy chọn, ***Context handler*** bao gồm **resource**   trong ***Context***
10. ***Context handler*** gửi các **attributes**  được yêu cầu và (tùy chọn) **resource**   tới ***PDP*** . Các ***PDP*** đánh giá ***policies*** .
11. Các ***PDP*** trả về phản ứng ***Context*** (bao gồm cả ***authorization decision (quyết định uỷ quyền)*** ) để ***Context handler***
12. ***Context handler*** dịch  ***Context*** đáp ứng sang định dạng phản hồi gốc của PEP . Trình xử lý ngữ cảnh trả về phản hồi cho PEP .
13. Các ***PEP*** đáp ứng các ***obligations***
14. Nếu ***access***  được cho phép, thì ***PEP*** cho phép ***access***  vào **resource**  ; nếu không, nó từ chối ***access***
15. **Thành phần của XACML**

Một XACML bao gồm 3 thành phần cơ bản sau:

* Rule
* Policy
* Policy Set



1. **Rule**:

Một Rule là đơn vị cơ bản nhất của Policy. Nó có thể tồn tại trong sự cô lập trong những tác nhân chính của miền XACML. Để trao đổi các Rule giữa các tác nhân chính, chúng phải được đóng gói trong Policy. Một Rule có thể được đánh giá dựa trên nội dung của nó. Các thành phần chính Của Rule:

* Một **target**.
* Một **effect**.
* Một **condition.**
* Biểu thức ***obligation***
* Biểu thức ***advice***
  1. **Target**
* Các **target** xác định tập hợp các yêu cầu mà các **Rule** được thiết kế để áp dụng trong các hình thức của một biểu thức logic trên ***các* attributes**  trong yêu cầu. Phần tử <Condition> có thể tinh chỉnh thêm khả năng ứng dụng được thiết lập bởi **target**. Nếu **Rule** được dự định áp dụng cho tất cả các thực thể của một loại dữ liệu cụ thể, thì thực thể tương ứng sẽ bị bỏ qua khỏi **target**. Một XACML ***PDP*** xác minh rằng các kết quả được xác định bởi ***target***  được thỏa mãn bởi các **attributes**  trong ***Context*** yêu cầu .
* Phần tử <Target> có thể không có trong <Rule> . Trong trường hợp này, **Target** của <Rule> giống với đích của phần tử <Policy>element.
  1. **Effect**
* Các ***effect***  của ***Rule*** chỉ ra ***Rule***-writer dự định hậu quả của một đánh giá "True" cho ***Rule*** . Hai giá trị được cho phép: " "Permit " và "Deny". Ngoài ra còn có hai các giá trị: “NotApplicable” và “Indeterminate”
  1. **Condition (*Điều kiện* )**
* ***Điều kiện*** biểu diễn một biểu thức Boolean để tinh chỉnh khả năng áp dụng của ***quy tắc*** ngoài các biến ***predicates*** ***(vị ngữ) được*** ngụ ý bởi ***mục tiêu*** của nó . Do đó, nó có thể vắng mặt.
  1. **Biểu thức *obligation*(nghĩa vụ)**
* ***Các*** biểu thức ***obligation*** có thể được thêm vào bởi người viết ***Rule***.
* Khi một ***PDP*** đánh giá một ***Rule*** chứa ***obligation*** biểu thức, nó đánh giá ***obligation*** biểu thức thành ***obligation*** và trả về một số những ***obligation*** với ***PEP*** trong ứng phó với ***bối cảnh*** .
  1. **Biểu thức *advice (tư vấn)***
* ***Các*** biểu thức ***advice*** có thể được thêm vào bởi người viết ***Rule*** .
* Khi một ***PDP*** đánh giá một ***Rule*** chứa ***advice*** biểu thức, nó đánh giá ***advice*** biểu thức hành ***advice*** và trả về một số những ***advice*** cho ***PEP*** trong ứng phó với ***Context*** .

1. **Policy**

Từ mô hình luồng dữ liệu, người ta có thể thấy rằng ***các Rule*** không được trao đổi giữa các thực thể hệ thống. Do đó, ***PAP*** kết hợp ***các Rule***  trong một ***Policy*** . Một **Policy** bao gồm bốn thành phần chính:

* Target
* A ***rule-combining algorithm***-identifier (thuật toán kết hợp thuật toán- quy tắc).
* set Rule
* Biểu thức ***obligation***
* Biểu thức ***advice***

1. **Mục tiêu chính sách (target policy)**

* Phần tử XACML <PolicySet> , <Policy> hoặc <Rule> chứa phần tử <Target> chỉ định tập hợp các yêu cầu mà nó áp dụng. Các <Target> một <PolicySet> hoặc <Policy> có thể được công bố bởi người viết của <PolicySet> hoặc <Policy> , hoặc nó có thể được tính từ <Target> yếu tố của <PolicySet> , <Policy> và <Rule> các phần tử chứa nó.
* Một thực thể hệ thống tính toán một <Target> theo cách này không được định nghĩa bởi XACML, nhưng có hai phương thức logic có thể được sử dụng. Trong một phương thức, phần tử <Target> của <PolicySet> hoặc <Policy> bên ngoài ("thành phần bên ngoài") được tính như là sự kết hợp của tất cả các phần tử <Target> của tham chiếu <PolicySet> , <Policy> hoặc < Rule > các phần tử ("thành phần bên trong").

1. **Rule-combining algorithm-identifier ( thuật toán kết hợp quy tắc)**

* Các ***thuật toán quy tắc kết hợp*** quy định cụ thể thủ tục theo đó các kết quả đánh giá các thành phần ***Rule*** được kết hợp khi đánh giá các ***Policy***, tức là ***decision*** (***quyết định)*** giá trị đặt trong phản ứng ***context***của ***PDP*** là giá trị của các ***Policy***, theo quy định của **thuật toán kết hợp quy tắc**. Một ***Policy*** có thể đã kết hợp các tham số ảnh hưởng đến hoạt động của ***thuật toán kết hợp quy tắc*** .

1. **Biểu thức Obligation (nghĩa vụ)**

* ***Các*** biểu thức **Obligation**  có thể được thêm vào bởi người viết ***policy***.
* Khi một ***PDP*** đánh giá một ***policy*** có chứa **Obligation** biểu thức, nó đánh giá **Obligation**  biểu thức thành **Obligation** và trả về một số những **Obligation** với ***PEP*** trong ứng phó với ***context***

1. **Biểu thức advice (tư vấn)**

* ***Các*** biểu thức ***advice*** có thể được thêm vào bởi tác giả của ***policy***.
* Khi một ***PDP*** đánh giá một ***policy*** có chứa ***advice*** biểu thức, nó đánh giá ***advice*** biểu thức thành ***advice*** và trả về một số những ***advice*** cho ***PEP*** trong ứng phó với ***context.***

1. **Policy set (bộ chính sách)**

Một ***Policy set*** bao gồm bốn thành phần chính:

* Target;
* A ***policy-combining algorithm***-identifier (***chính sách kết hợp thuật toán*** –định danh)
* A policy set;
* Biểu thức Obligation
* Biểu thức***advice***

1. **Policy-combining algorithm (Thuật toán kết hợp chính sách)**

* ***Các Policy-combining algorithm quy định cụ thể thủ tục theo đó các kết quả đánh giá các thành*** phần  ***Policy*** được kết hợp khi đánh giá ***Policy set***, tức là các Quyết định giá trị đặt trong phản ứng ***context*** của ***PDP*** là kết quả của việc đánh giá ***policy set*** , theo quy định của **Policy-combining algorithm** . Một ***Policy set*** có thể kết hợp các tham số ảnh hưởng đến hoạt động của **Policy-combining algorithm**

1. **Biểu thức Obligation (nghĩa vụ)**

* Tác giả của ***Policy set***  có thể thêm các biểu thức ***obligation*** vào ***Policy set*** ngoài các biểu thức có trong các ***Rule***  thành phần , ***các Policy*** và ***các Policy set***.
* Khi một ***PDP*** đánh giá một ***Policy set***.có chứa ***các* Obligation** biểu thức, nó đánh giá **Obligation** biểu thức thành **Obligation** và trả về một số những **Obligation** với ***PEP*** trong phản ứng của nó ***Context***

1. **Biểu thức Advice**

* ***Các*** biểu thức ***Advice*** có thể được thêm vào bởi người viết của ***Policy set***..
* Khi một ***PDP*** đánh giá một ***bộ sách*** chứa ***Advice*** biểu thức, nó đánh giá ***Advice*** biểu thức thành ***Advice*** và trả về một số những ***Advice***  cho ***PEP*** trong ứng phó với ***Context***

1. **Logic Vị từ**
2. **The Logic of XACML**
   1. **Xacml**

Việc đánh giá các thành phần XACML bắt đầu từ đánh giá hàng loạt và nó được tiếp tục lặp đi lặp lại cho đến khi đánh giá **PolicySet**. Các giá trị **Match**, **AllOf**, **AnyOf** và **Target** đều khớp nhau, không khớp hoặc không xác định. Giá trị có thể không xác định nếu có lỗi trong quá trình đánh giá để không thể đưa ra quyết định tại thời điểm đó. Việc đánh giá **Rule** phụ thuộc vào Đánh giá **Target** và Đánh giá **Condition**. Thành phần **Condition** là một tập hợp các công thức mệnh đề mà mỗi công thức được đánh giá là đúng, sai hoặc không xác định. Điều kiện trống luôn được đánh giá là đúng. Kết quả của **Rule** là có thể áp dụng, không áp dụng hoặc không xác định. **Rule** áp dụng có hiệu lực hoặc từ chối hoặc cho phép. Cuối cùng, việc đánh giá **Policy** và **PolicySet** được dựa trên thuật toán kết hợp mà kết quả có thể được áp dụng (với hiệu ứng của nó hoặc từ chối hoặc cho phép), không áp dụng hoặc không xác định.

* 1. **Three-Valued Lattice (Lưới ba giá trị)**

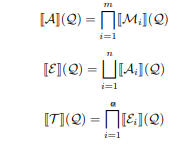
Chúng tôi sử dụng logic có giá trị ba để xác định giá trị đánh giá XACML. Chúng tôi định nghĩa L3 =(V3, ≤ )là mạng ba giá trị trong đó V3 là tập hợp {>, I, ⊥} và ⊥ ≤ I ≤ T.

Chúng tôi sử dụng ký hiệu J.K để ánh xạ các phần tử XACML vào các giá trị đánh giá của chúng. Các đánh giá các thành phần XACML với các giá trị trong V3 được tóm tắt trong Bảng

|  |  |  |  |
| --- | --- | --- | --- |
| V3 | Match and Target value | Condition value | Rule, Policy and PolicySetvalue |
| T | Match | True | Applicable (either deny or permit) |
| ⊥ | not match | false | not applicable |
| I | indeterminate | indeterminate | indeterminate |

* 1. **The Match Evaluation**

Để M là một trận đấu, A = M1 ∧. . . ∧ Mm là một AllOf, E = A1 ∨. . . ∨ An là một AnyOf, T = E1 ∧. . . ∧ Eo là mục tiêu và Q là yêu cầu. Sau đó, các đánh giá của AllOf, AnyOf và Target như sau:

  
Tóm lại, chúng ta có thể đơn giản hóa việc đánh giá mục tiêu như sau:



* 1. **Condition Evaluation**

Chúng tôi xác định hàm đánh giá có điều kiện eval là một hàm tùy ý để đánh giá

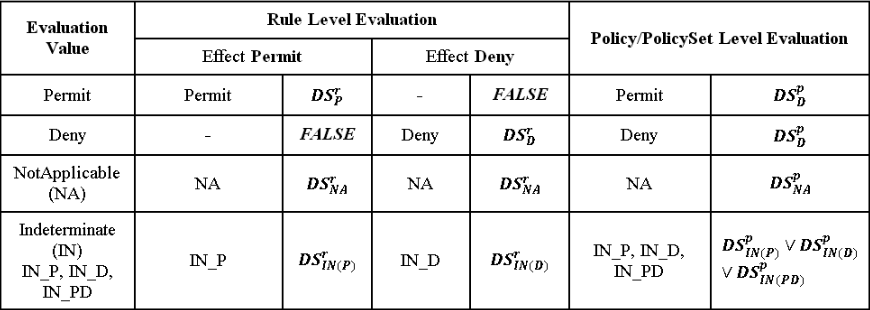
Điều kiện để giá trị trong V3 cho một thành phần Yêu cầu Q. Đánh giá

Điều kiện được định nghĩa như sau:



* 1. **Extended Values**

Chúng được biểu diễn qua bảng sau:



DS: biểu thức

R: rule

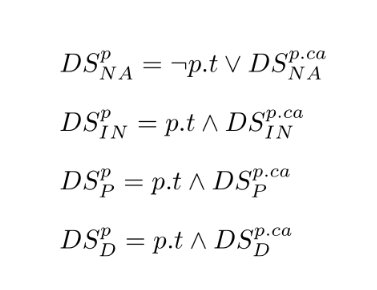
P: Permit: Cho phép

D: Deny: Từ chối

NA: Not Applicable: Không áp dụng

IN: Indeterminate: Không xác định

* 1. **Biểu thức rút gọn**

 Được biểu diễn qua hình sau:

**Chú thích:**

p.t: là Target

p: là Permit (cho phép)

in: là Indeterminate (erro)

Na: là NotApplicable (không liên qua)

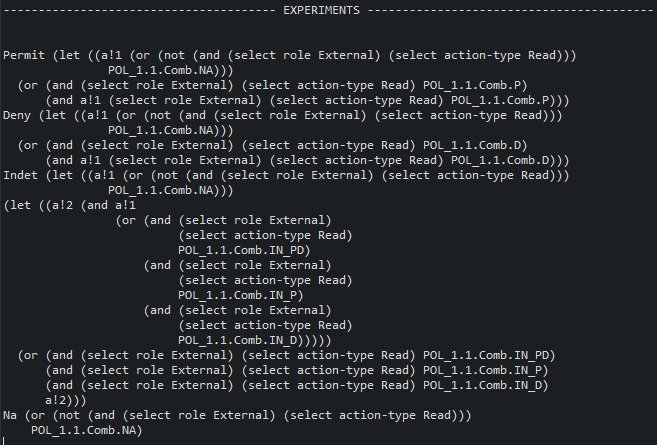
D: là Deny ( không cho phép)

**p.ca:** làpolicyCombining Algorithm

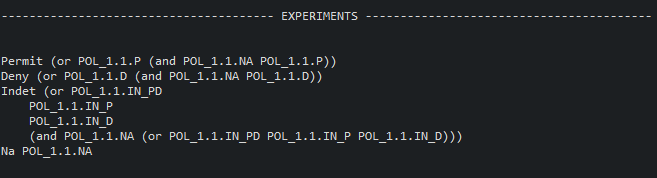
1. **Tìm hiểu cụ thể Policy và cách thu gọn**

[**QueryTest.java**](file:///F:\Documents%20Programming\Java\Project\research_group\source\XACMLSMT.jar.src\src\java\experiments\QueryTest.java)

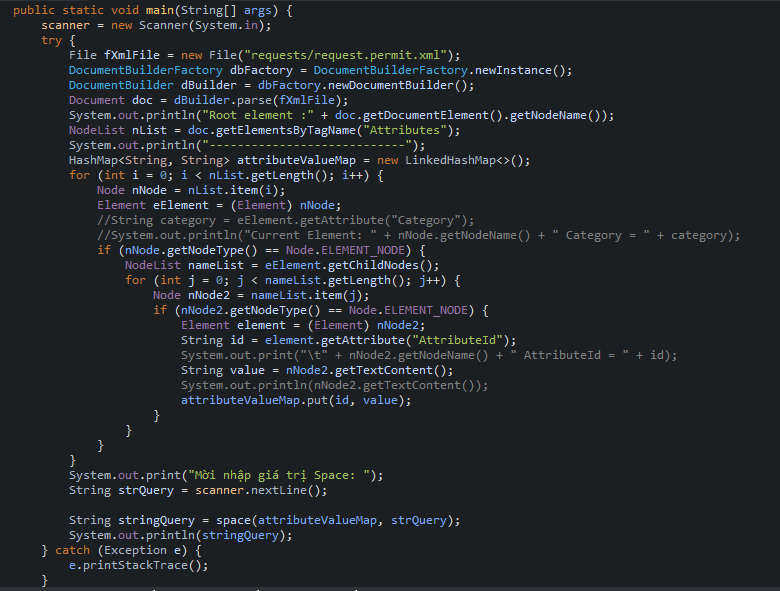
SHOW\_AT\_POLICY\_WITH\_TARGET\_COMBINATION



SHOW\_AT\_POLICY\_ONLY

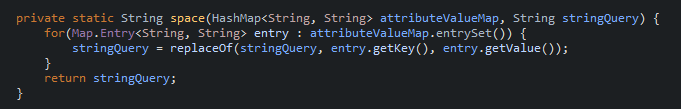


1. **Convert Request.permit.xml to stringQuery**



1. **Write Function Space(**hash map {string, string}), stringQuery)

stringQuery: p\_1 or d\_1 or na\_1 or in\_1



1. Kỹ thuật viết XML bằng ALFA