Nutzerorientierte Gestaltung eines Access Control Policy Management Interfaces

Sascha Wagner[1], Dominic Heutelbeck[2]
Lehrgebiet Multimedia und Internetanwendungen, Fakultät für Mathematik und Informatik, Fernuniversität Hagen[1]
FTK e.V. Forschungsinstitut für Telekommunikation und Kooperation[2]

Zusammenfassung

Dieser Beitrag stellt die Konzeption eines nutzerorientierten Access Control Policy Management Interfaces zur Erstellung und Bearbeitung von Autorisierungsrichtlinien beruhend auf dem beziehungsbasierten Zugriffsmodell „Structure and Agency Based Access Control“ (STABAC) dar. Der Schwerpunkt dieser Arbeit liegt in der Darstellung der eingesetzten Vorgehensweise, der vorhandenen Herausforderungen sowie der erzielten Werkzeuge und Lösungsmustern.

1 Einleitung

Kooperationen und Kollaborationen sind im Arbeitsumfeld allgegenwärtig. Ein wichtiger Aspekt für eine vertrauensvolle Zusammenarbeit stellt die Sicherheit der ausgetauschten und der zur Verfügung gestellten Informationen und Daten dar. Speziell in verteilten Systemen steigt der Bedarf an dynamischen Änderungen der Autorisierungsrichtlinien (engl. Access Control Policies). Neben klassischen Zugriffsmodellen wie Role-Based Access Control (Ferraiolo & Kuhn 1992; Sandhu et al. 1995) oder Attribute-Based Access Control (Kolter et al. 2007) haben sich ausgehend von der Verbreitung von Online Social Networks (wie bspw. Facebook) Modelle entwickelt, welche u.a. die Beziehungen der Anwender und Ressourcen zur Beurteilung von Zugriffsanfragen einbeziehen. Diese Modelle werden u.a. als Relationship-Based Access Control Models (ReBAC) bezeichnet (Gates 2007). Mit Hilfe solcher Modelle können komplexe und dynamische Autorisierungsrichtlinien (Access Control Policies) erstellt werden. Eine Herausforderung ist jedoch die gebrauchstaugliche Erstellung von Autorisierungsrichtlinien. Speziell im Bereich von ABAC (Attribute-Based Access Control) und ReBAC existieren zahlreiche technisch orientierte Veröffentlichungen, wie beziehungsbasierte Zugriffskontrolle vollzogen werden kann. Konkrete Darstellungen, wie Anwender im Rahmen der Erstellung von Autorisierungsrichtlinien unterstützt werden können, fehlen jedoch zu meist (Zain et al. 2015).

Veröffentlicht durch die Gesellschaft für Informatik e.V. 2016 in
B. Weyers, A. Dittmar (Hrsg.):
Mensch und Computer 2016 – Workshopbeiträge, 4. - 7. September 2016, Aachen.

http://dx.doi.org/10.18420/muc2016-ws03-0003

Der vorliegende Beitrag beschreibt eine nutzerorientierte Vorgehensweise zur Erstellung eines ersten Access Control Policy Management Interfaces zur Erstellung und Bearbeitung von Zugriffsrichtlinien auf Basis des sich in der Entwicklung befindlichen beziehungsbasierten Zugriffsmodells „Structure and Agency Based Access Control“ (STABAC)[1]. Im EU Forschungsprojekt Smart Vortex[2] wurden erste Erkenntnissen über die Tragfähigkeit und das Potenzial des STABAC-Ansatzes gewonnen (SMART VORTEX 2016). Es wurde ebenfalls festgestellt, dass Anwender bei der Erstellung von Autorisierungsrichtlinien unterstützt werden müssen.

2 Hintergrund und verwandte Arbeiten

Die Erstellung von komplexen Autorisierungsrichtlinien kann eine herausfordernde Aufgabe für Anwender sein. Für die Erstellung stehen technische Beschreibungssprachen wie beispielsweise die Authorization Specification Language (ASL), Ponder oder die eXtensible Access Control Markup Language (XACML) zur Verfügung. Nach Conti et al. weisen diese jedoch eine unzureichende Gebrauchstauglichkeit auf (Conti et al. 2014). Eine Autorisierungsrichtlinie, welche auf Papier kompakt ausgedrückt werden kann, kann bspw. in XACML zu einem umfangreichen Dokument mit zahlreichen XML-Tags und vielfältigen schwer zu deutenden Funktionen werden. Dieser Umstand kann zu einer fehlerhaften Erstellung sowie zu einer schwierigen Deutung der Auswirkung einer Autorisierungsrichtlinie führen (Nergaard et al. 2015). Um Autorisierungsrichtlinien benutzerfreundlich erstellen zu können, wurden verschiedene Editoren entwickelt (Wagner & Heutelbeck 2015). Diese lassen sich in textbasierte und grafikbasierte unterteilen. Laut Stepien et al. haben diese meist das Ziel die Beschreibungssprache transparent darzustellen und den Lernaufwand zu reduzieren. Hierbei wird u.a. auf Dropdown-Listen zur Auswahl der grammatikalischen Elemente gesetzt (Stepien et al. 2014).

3 Herausforderungen und Ziel

Eine Schwierigkeit bei der Entwicklung des Interfaces besteht in der Neuartigkeit des eingesetzten Zugriffsmodells (STABAC) sowie dem Umstand, dass derzeit kein Interface gefunden werden konnte, welches die Erstellung von beziehungsbasierten Autorisierungsrichtlinien durch ein explizites User Interface unterstützt.

Das Zugriffsmodell STABAC nutzt Informationen aus einem semantischen Graphen, welcher u.a. Personen und Ressourcen einer Organisation abbildet, um diese im Rahmen der Zugriffskontrolle zu nutzen. Personen, Ressourcen oder auch Gruppen werden dabei als

[1] Eine ausführliche Beschreibung des STABAC-Modells befindet sich derzeit im Veröffentlichungsprozess.

[2] Siehe: http://www.smartvortex.eu/

Knoten und ihrer Beziehungen zueinander als Kanten abgebildet. Diese Informationen können im Rahmen von Zugriffsentscheidungsanfragen berücksichtigt werden.

Eine Idee des STABAC-Ansatzes ist, dass es wiederkehrende Beziehungsstrukturen gibt, welche im Rahmen von Autorisierungsrichtlinien durch den Anwender parametrisiert werden können (siehe Abbildung 1). Diese Muster werden im Rahmen dieses Beitrags als Strukturmuster bezeichnet.

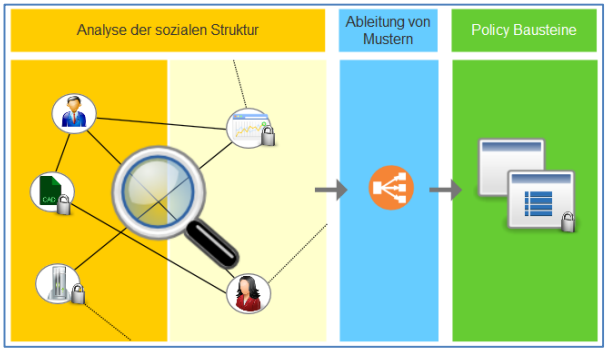

Abbildung 1: Vereinfachte Darstellung des STABAC-Ansatzes

Ziel der Entwicklung ist die Erschaffung eines ersten Open Source Interfaces für das beziehungsbasierte Zugriffsmodell STABAC. Das aktuell erzielte Ergebnis ist als Momentaufnahme zu verstehen, welches als Evaluationsobjekt für zukünftige Anwenderstudien und für andere Wissenschaftler als Informationsquelle dienen soll.

4 Design des Editors

Im Rahmen der Entwicklung wurde eine iterative nutzerorientierte Vorgehensweise angewendet, welche sich an den Lean UX Prinzipien: LEARN – BUILD – MEASURE[3] orientiert.

Die Entwicklung des Editors folgte den folgenden Schritten:

1. Ableitung von generellen UI-Prinzipien aus bestehenden Guidelines und Heuristiken (siehe (Wagner & Heutelbeck 2015))
2. Festlegung auf die primäre Zielgruppe der Administratoren, Erstellung von Personas und Modellierung von Anwendungsfällen[4]

[3] Siehe: https://www.interaction-design.org/literature/article/a-simple-introduction-to-lean-ux

[4] Basierend auf dem Ergebnis des EU Forschungsprojektes Smart Vortex.

3. Ausarbeitung von verschiedenen Lösungsmöglichkeiten sowie erlebbar machen von Lösungsideen durch Rapid Prototyping (interaktive HTML-Prototypen)
4. Bewertung der Lösungsmöglichkeiten im Kernprojektteam bestehend aus Software-, Usability- und Security Experten
5. Einarbeitung des generierten Feedbacks

5 Der Editor

Das entwickelte Interface setzt auf die Analyse bestehender Lösungen und Arbeiten sowie den daraus erkennbaren Lösungsansätzen, wie: *Natürlich-sprachliche Texteingabe, alternative Syntax, Templates, strukturierte Eingaben, Reduzierung des Funktionsumfangs und visuelle Unterstützung* (siehe (Wagner & Heutelbeck 2015)).

Basierend auf den im Kapitel 4 beschriebenen Schritten, wurde anschließend ein Software-Prototyp entwickelt und eine Testumgebung geschaffen, welche im weiteren Verlauf als Evaluationsumgebung für Anwenderstudien dienen soll. Das Interface wurde dabei in einem XACML-Umfeld in den WSO2 Identity Server[5] integriert.

Um die UI-Komplexität zu verringern wurde die Oberfläche in drei aufeinander aufbauenden Schritten unterteilt:

1. Festlegung für welches Objekt die Autorisierungsrichtlinie gelten soll (*For...)*
2. Definition, welche Aktionen bei welchem Strukturmuster ausgeführt werden sollen (*do...*)
3. Anzeige der Autorisierungsrichtlinie in einer kompakten textuellen Repräsentation (*Your policy...*) [6]

Das Interface setzt zudem auf etablierte UI-Pattern, wie Details-On-Demand, Progressive Disclosure, Auto Completion und Hilfetexte.

Nachfolgend werden die wesentlichsten Punkte des UI-Konzeptes kompakt vorgestellt.

Im ersten Schritt (*1. For...*) definiert der Anwender für welche Ressource die Richtlinie gelten soll.

[5] Siehe: http://wso2.com/products/identity-server/

[6] Dieser Punkt befindet sich derzeit in Bearbeitung.

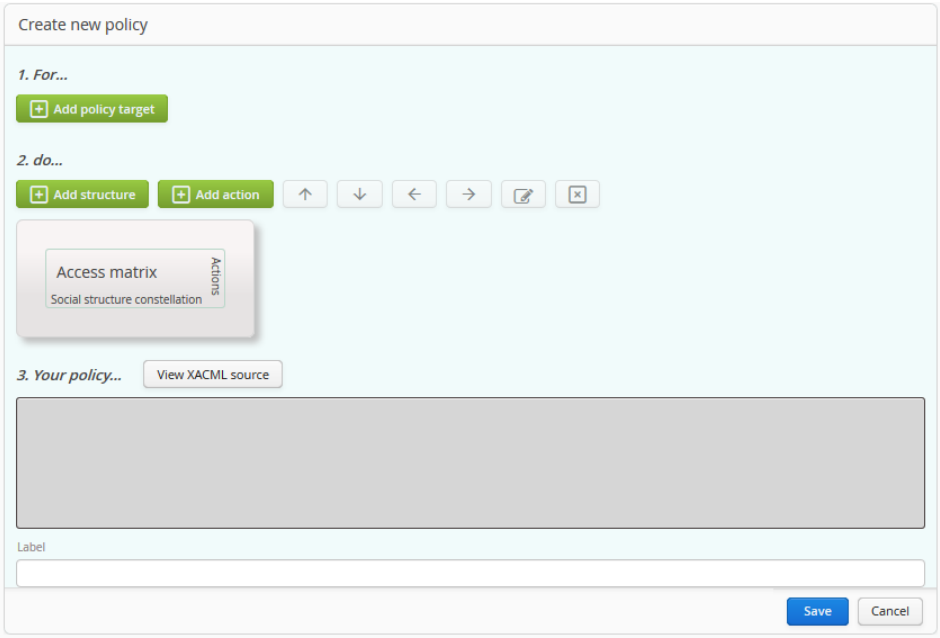

Abbildung 2: Übersicht des STABAC-Editor

Im zweiten Schritt (*2. do...*) erfolgt die Darstellung einer Access Matrix, angelehnt an klassische Access Control Lists (ACLs). In dieser kann der Anwender u.a. Regeln festlegen, welche Aktionen bei welcher Konstellation der Strukturmuster gelten sollen. Die Festlegung wird durch eine visuelle Repräsentation des Strukturmusters (siehe Abbildung 3) und zusätzliche Erklärungen unterstützt. Im weiteren Projektverlauf ist eine skalierbare Version der Access Matrix angedacht, welche sich an dem Visual Information-Seeking Mantra von Shneiderman[7] (Overview first, zoom and filter, then details-on-demand) orientiert.

[7] Siehe http://www.ifp.illinois.edu/nabhcs/abstracts/shneiderman.html

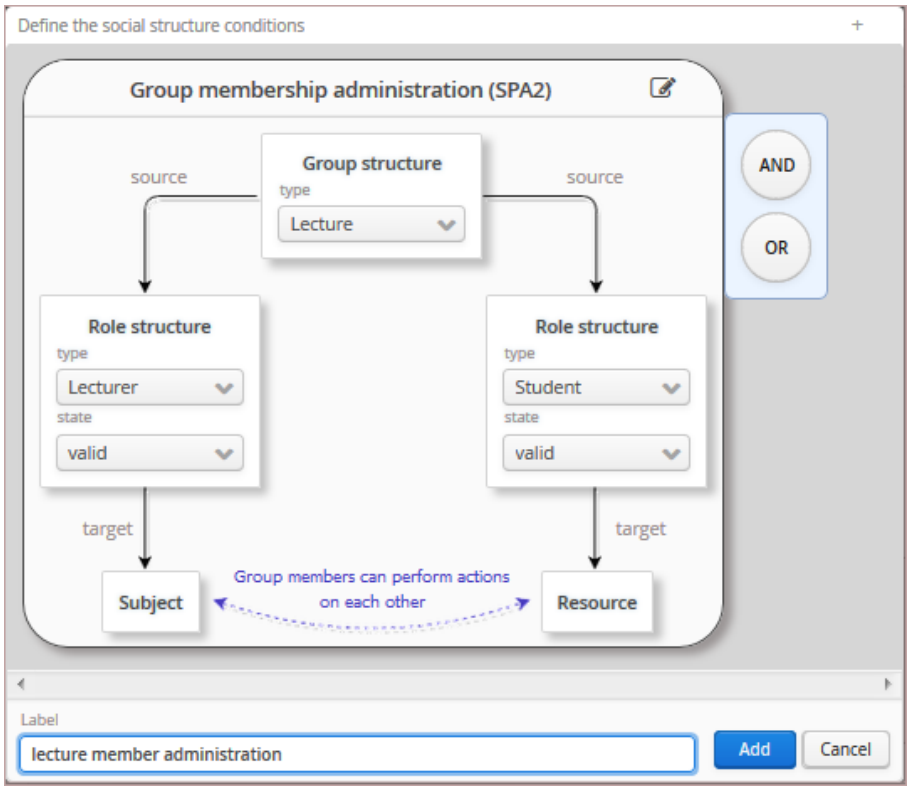

Abbildung 3: Parametrisierung eines Strukturmusters

Der letzte Schritt (*3. Your policy...*) soll dem Anwender zukünftig eine textuelle und kompakte Repräsentation seiner Autorisierungsrichtlinie vermitteln (siehe Seite 5: Lösungsansätze). Dieser Bereich befindet sich derzeit noch in einem frühen unausgereiften Konzeptions- und Entwicklungsstand. Daher ist der in Abbildung 4 dargestellte Inhalt nicht repräsentativ.

Durch den dargestellten schrittweisen Aufbau der Oberfläche soll Transparenz über den Erstellungsfortschritt sowie abschließend eine schnelle Übersicht über die wichtigsten Informationen der Autorisierungsrichtlinie auf einer Seite gewährleistet werden (siehe Abbildung 4). Um die Komplexität zu verringern, wurde dabei bewusst auf die Anzeige detailreicher Informationen verzichtet. Diese stehen dem Anwender durch den Einsatz des UI-Pattern „Details-On-Demand“ zur Verfügung.

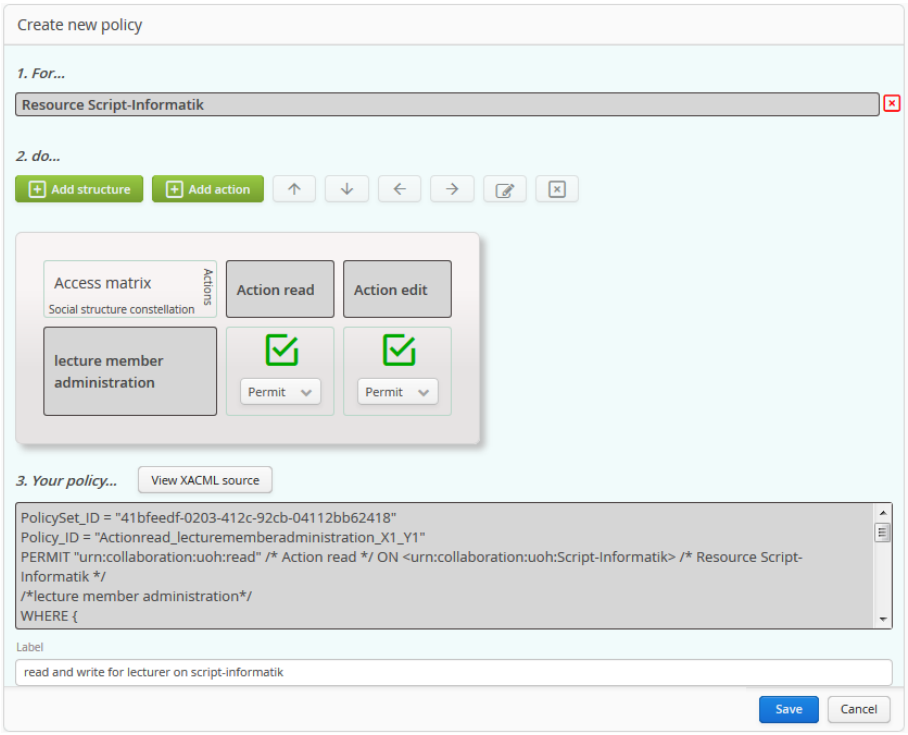

Abbildung 4: Darstellung einer erstellten Autorisierungsrichtlinie

6 Zusammenfassung und Ausblick

Der vorliegende Beitrag stellt ein erstes nutzerorientiertes UI-Konzept für einen beziehungsbasierten Editor zur Erstellung und Bearbeitung von Autorisierungsrichtlinien basierend auf dem Zugriffsmodell STABAC vor. Der entstandene Editor führt den Anwender (aktuell Administratoren) Schritt für Schritt durch den Entstehungsprozess einer Autorisierungsrichtlinie. Kernelement ist hierbei das Festlegen, unter welcher Konstellation, welche Aktionen durchgeführt werden sollen. Zu diesem Zweck bietet die Oberfläche parametrisierbare visuelle Strukturmuster, welche dem Anwender eine schnelle Konfiguration der Konstellation ermöglichen soll. Um eine Übersicht über alle Regeln einer Autorisierungsrichtlinie zu ermöglichen wird eine Access Matrix eingesetzt. Diese nutzt u.a. Farbcodierung und Text, um einen schnellen Überblick über den jeweiligen Zustand (z.B. „Allow“, „Deny“, „N/A“) zu kommunizieren. Zusätzlich soll die Autorisierungsrichtlinie zukünftig textuell, orientiert an natürlicher Sprache, dargestellt werden.

Der aktuelle Entwicklungsstand des Editors dient in erster Linie als Evaluationsobjekt für die im nächsten Schritt stattfindende formative Usability Evaluation mit Anwendern und als Informationsquelle für andere Wissenschaftler. In der geplanten Usability Evaluation sollen qualitative und quantitative Daten gesammelt werden, um Optimierungspotenziale zur

Verbesserung des aktuellen Standes zu ermitteln. Des Weiteren wird an einer Vereinfachung der Darstellung der Strukturmuster, eine skalierbare Access Matrix sowie an einer Verringerung des technischen Vokabulars gearbeitet.

Literaturverzeichnis

Conti, R., Matteucci, I., Mori, P., & Petrocchi, M. (2014). An Expertise-Driven Authoring Tool for E-Health Data Policies. *2014 IEEE 27th International Symposium on Computer-Based Medical Systems (CBMS)*, 82-87.

Ferraiolo, D., & Kuhn, D. (1992). Role-Based Access Controls. *15th National Computer Security Conference* , (S. 554-563). Baltimore MD.

Gates, C. E. (2007). Access Control Requirements for Web 2.0 Security and Privacy. *Proceedings of the Web 2.0 security & privacy 2007 workshop.*

Kolter, J., Schillinger, R., & Pernul, G. (2007). A Privacy-enhanced Attribute-based Access Control System. *DBSec 2007.* Springer.

Nergaard, H., Ulltveit-Moe, N., & Gjøsæter, T. (2015). A Scratch-based Graphical Policy Editor for XACML. *Olivier Camp (Hg.): Proceedings of the 1st International Conference on Information Systems Security and Privacy*, 182-190.

Sandhu, R., Coyne, E., Feinstein, H., & Youman, C. (1995). Role-Based Access Control Models. *IEEE Computer*, *29*(2), 38-47.

SMART VORTEX Project Homepage. (2016). Abgerufen am 01. 06 2016 von http://www.smartvortex.eu/

Stepien, B., Felty, A., & Matwin, S. (2014). A non-technical XACML target editor for dynamic access control systems. *Waleed W. Smari (Hg.): 2014 International Conference on Collaboration Technologies and Systems (CTS 2014)*, 150-157.

Wagner, S., & Heutelbeck, D. (2015). Usable-Policy-Management in kollaborativen Szenarien - Herausforderungen und Chancen. *Mensch und Computer 2015 – Workshopband* (S. 657-665). Berlin: De Gruyter Oldenbourg.

Zain, S., Rizvi, R., Fong, P. W., Crampton, J., & Sellwood, J. (2015). Relationship-Based Access Control for an OpenMRS. In ACM (Hrsg.), *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies*, (S. 113-124). Vienna, Austria.

Kontaktinformationen

Sascha Wagner (sascha.wagner@fernuni-hagen.de)
Dominic Heutelbeck (dheutelbeck@ftk.de)