# A Systematic Literature Review on Extensions to Role-Based Access Control

Eric D. Helms, JeeHyun Hwang, Laurie Williams, Tao Xie

*North Carolina State University*

*Department of Computer Science*

*890 Oval Drive, Box 8206*

*Raleigh, NC 27695-2858*

**Abstract**

**Context**: Since the National Institute of Standards and Technology (NIST) proposed Role-Based Access Control (RBAC) standard using RBAC reference models (RBAC$_m$ in short) in the late 1990's, computing in new domains has led to proposals for extensions to the RBAC$_m$; for example adding context around permission granting and role activation.

**Aim**: The goal of our work is to aid practitioners and researchers in choosing an extension to the RBAC$_m$, and in understanding how extensions to RBAC$_m$ are evaluated by providing an assessment of the state of extensions to the RBAC$_m$. We accomplished this through: establishment of a set of extension categories, an examination of the state of the art in evaluations of extension models, and a breakdown of the motivations that have led to extension of the RBAC$_m$.

**Method**: We performed a systematic literature review that yielded 1,716 papers, of which 28 were deemed primary sources for inclusion as extensions to the RBAC$_m$.

**Results**: Our results show that extensions to the RBAC$_m$ can be classified under eight categories: Constraint, Context, Organization, Privacy, Task, Spatio-Temporal, Spatial, and Temporal. We identified only eight of the 28 papers provided an implementation of their model in the form of an enterprise application or prototype. We found that the primary domains that inspired extensions were the medical domain with nine of the 28 models, enterprise workflows with five of 28, and mobile computing with five of 28.

**Conclusions**: When examined as an entire group, the eight categories can be aligned under the single category of context. Our literature review shows that the state of evaluation of RBAC model evaluation is limited with one paper doing an evaluation comparing itself to the RBAC$_m$, eight presenting example scenarios of their model in action, and 12 of the 28 models lacking an evaluation that we could discern. The landscape of extensions to the RBAC$_m$ serves as a basis from which a new, and vetted, extended RBAC reference models should be derived.

*Keywords:* RBAC, access control, systematic literature review

## 1. Introduction

Role-Based Access Control (RBAC) is widely used for maintaining and managing an organization's access control based on assigning permissions to roles, and roles to users, instead of assigning the permissions directly to individual users. RBAC is used for securing various applications including web services, database applications, and healthcare applications. In RBAC, roles represent a set of permissions needed to perform a particular job function within an organization. Multiple users who are involved in that specific job function within the organization can then be assigned to a single role to inherit the required access. The ability to logically group users into roles associated with permissions becomes paramount for managing access control as an organization grows and the number of permissions and users scales upward. As permissions can be managed by a role instead of a user, RBAC has an advantage to significantly reduce complexity of security administration. For example, if a user requires access to resources associated with a manager role within a given organization, security policy administrators need only associate the user with the manager role instead of assigning permissions of the manager role to the user.

The use of RBAC has become popular since the National Institute for Standards and Technology (NIST) first proposed RBAC standard in the 1990s [1]. The RBAC standard was first proposed when NIST requested that a unified standard be created by combining the Ferraiolo and Kuhn model [2] with the framework proposed by Sandhu, et al [3]. In 2004, this standard was adopted as ANSI/INCITS 359-2004 approved by American National Standards Institute (ANSI) and the InterNational Committee for Information Technology Standards (INCITS). The development of a standard was inspired by an economic impact study done during the 1990s and later confirmed in 2010[1] that showed the cost savings of RBAC implementation and maintenance. Prior to the development of the RBAC standard, vendors proposed and implemented their own defined RBAC features without general agreement on a unified definition of RBAC or feature set. The RBAC standard includes RBAC reference models ($\text{RBAC}_m$ in short). $\text{RBAC}_m$ serve as a basis for defining scope of RBAC features and functional specifications of RBAC features.

Innovation and the use of RBAC in new domains has led to scenarios that, by some accounts, the $\text{RBAC}_m$ cannot handle [4]. For example, Ni et al. [5] state that $\text{RBAC}_m$ are "not designed to enforce privacy policies and barely meet privacy protection requirements" with the introduction of privacy concerns in domains such as healthcare. Since the introduction of the RBAC standard, extension models to the $\text{RBAC}_m$ have appeared in the literature, each adding one or more features on top of components in the $\text{RBAC}_m$. Further, these extension models are each building upon and adding features to a standard that was designed to reduce the economic impact experienced by enterprises and increase interoperability [1].

*The goal of our work is to aid practitioners and researchers in choosing an extension to the $\text{RBAC}_m$, and in understanding how extensions to $\text{RBAC}_m$ are evaluated by providing an assessment of RBAC extensions. We accomplished this through: establishment of a set of RBAC extension*

[1] `http://csrc.nist.gov/groups/SNS/rbac/documents/20101219_RBAC2_Final_Report.pdf`

*categories, an examination of the state of the art in evaluations of RBAC extension, and a breakdown of the motivations that have led to extensions to RBAC$_m$.* To accomplish this goal, we seek to answer following research questions:

- RQ1: What categories exist within extensions to RBAC$_m$?
- RQ2: What are the motivations behind extensions to RBAC$_m$?
- RQ3: Do the extensions to RBAC$_m$ have corresponding implementations?
- RQ4: How are extensions to RBAC$_m$ evaluated theoretically and in practice?
- RQ5: What domains have extensions to RBAC$_m$ been created for?
- RQ6: What commonalities or generalizations exist across all categories?

We performed a systematic literature review designed to explore the current body of research in the area of extensions to the RBAC$_m$. The review we performed yielded 1,716 papers, of which 28 were deemed primary sources for inclusion as extension models to the RBAC$_m$. This review is intended to serve as a starting place for researchers looking to tackle new problems in the realm of authorization and as a reference guide for discovering the current state of extensions to the RBAC$_m$. For developers looking to find a model to fit their potentially unique access control needs, this review provides a basis for comparison and easy look up for what extension model to use. Furthermore, the review provides insight into the state of the art in evaluating RBAC extension models.

Our research provides contributions to the community in following ways:

- Summarizes current research on extensions to RBAC$_m$.
- Guides a direction for a new standard based on an extension to the RBAC$_m$.

The rest of the paper is organized as follows. Section 2 presents background and the RBAC standard. Section 3 presents methodology and process, which we used in conducting the systemic literature review. Sections 4-9 present analysis and discussion of the research questions. Section 10 discusses issues about extensions to RBAC$_m$. Section 11 concludes the paper.

## 2. Role-Based Access Control Standard

RBAC provides effective and efficient permissions management for operations, especially when sharing resources within an organization. Prior to the creation of the NIST RBAC standard, no general agreement on the definition of RBAC existed among practitioners or within the research community. Without a unified definition of RBAC, software developers described similar concepts and features of RBAC models using different terminology. This lack of consistent terminology was shown to slow the implementation of RBAC [1]. Moreover, in cases where organizations were concerned with adopting RBAC, evaluation and comparison of RBAC technologies developed by different vendors was difficult. NIST, in collaboration with industry and academics, worked on defining a set of consensus RBAC concepts and terminology and proposed a standard

for RBAC that addressed these cost and interoperability issues by developing a common definition that can be used across different vendors.

NIST's work can directly benefit organizations by lowering cost of early phase Research and Development, and implementation of RBAC. Since the RBAC standard was first introduced, a 2010 report by RTI International showed that the the rate of RBAC adoption has rapidly grown over recent years [1]. The analysts estimate that by the end of 2010 at least a portion of permission of systems will use RBAC for more than half of users at organizations with more than 500 employees. The analysts estimate that RBAC technology has generated $6.1 billion in net economic benefits to industry where NIST RBAC standard work saved $1.1 billion.

The NIST RBAC standard proposed by Ferraiolo et al. [6] and later adopted as the official standard for RBAC by the INCITS includes three components of RBAC: core RBAC, hierarchical RBAC, and constrained RBAC. Each component includes a RBAC reference model. RBAC$_m$ includes RBAC reference models of these three components. We first describe the core RBAC, associated entities and other terminology encountered across the space of our review. We next describe hierarchical RBAC and constrained RBAC, which are developed by incorporating new features into the core RBAC.

### *2.1. Core RBAC*

The four entities of the core RBAC reference model are:

- a set of *Users*: A user can be a person or an agent.
- a set of *Roles*: A role is a collection of permissions to perform a specific job function in an organization.
- a set of *Permissions*: A permission refers to an access mode that can be exercised on an object in the system and a session relates a user to possibly many roles.
- a set of *Sessions*: In each session, a user can be assigned to some of the roles, only when the corresponding role is enabled for activation for that time.

In RBAC, a user can exercise a permission only if the user is assigned to a role. In addition to the four basic entities, two functions are defined: user assignment ($UA$) and permission assignment ($PA$) functions. $UA$ assignment of users to roles. $PA$ assignment of permissions to roles.

Figure 1: Diagram in Core RBAC reference model[2].

Figure 1 presents an overview of core RBAC reference model diagram where entities and their relations are described. Let "USERS", "ROLES", "OBS", "OPS", "PRM", and "SESSIONS" denote users, roles, objects, operations, permissions, and sessions, respectively. Permissions are associated with possible users' pre-defined operation on an object (e.g., execute a file). Note that, at user or role activation, a session associated with user or role is established.

### *2.2. Hierarchical RBAC*

The hierarchical RBAC reference model adds role hierarchies ($RH$) feature to the core RBAC reference model. $RH$ incorporates a structure of roles in an organization using inheritance relationships among attributes such as roles. The role structure in an organization may use a role $r_1$, which inherits all permissions of another role $r_2$. For example, a manager role may inherit all permissions of a employee role. Role hierarchies help simplify access control policy creation and maintenance by reducing the number of individual role assignments for a user. Formally, the role inheritance relation is shown as $RH \subseteq ROLES \times ROLES$ describing the many-to-many mapping role inheritance relation. General role hierarchies can be extended to use the concept of multiple inheritances where $r_1$ inherits all permissions from more than one roles.

### *2.3. Constrained RBAC*

The constrained RBAC reference model incorporates separation of duty relations to the core or the hierarchical RBAC reference. Separation of duty relations enforces conflict of interest among roles. This model defines two types of separation of duty relations; static and dynamic.

- Static Separation of Duty (SSoD): SSoD restricts the conflicting-roles, which can be assigned to a single user statically. Consider that roles $Role_A$ and $Role_B$ are conflicting with each other. On situations where multiple roles can be associated with a single user, no permission is given to a user who is assigned to both $Role_A$ and $Role_B$ statically. SSoD is known to be too rigid for practical use in cases where a user should have permissions when a user is assigned to both $Role_A$ and $Role_B$.

- Dynamic Separation of Duty (DSD): Dynamic SoD is known to be more flexible than SSoD. DSD restricts the conflicting-role assignments dynamically that are associated with a user. Consider that roles $Role_A$ and $Role_B$ are conflicting with each other. For the situations where multiple roles can be associated with a single user, no permission is given to a user who is assigned to both $Role_A$ and $Role_B$ dynamically.

## 3. Methodology and Process

We adopted and applied a systematic literature review following recommendations from Kitchenham's suggested processes [7]. The systematic literature review process was broken down into four stages and the rest of this section is broken down by each stage. The stages were:

- Step 1: Development of a search strategy
- Step 2: Elimination of papers based on title criteria
- Step 3: Elimination of papers based on abstract criteria
- Step 4: Elimination of papers based on content and matching to elimination criteria

Table 1: Paper counts after applying search strategy

| | **RBAC** | **role based access control** | **role-based access control** | **Total** |
|---|---|---|---|---|
| Google Scholar | 651 | 213 | 435 | 1299 |
| ACM Portal | 500 | 20 | 720 | 1240 |
| IEEExplore | 200 | 40 | 230 | 470 |
| CiteSeerX | 100 | 100 | 150 | 350 |
| | | | | |
| Totals | 1451 | 373 | 1535 | 3359 |
| Combined | | | | **1716** |

### *3.1. Step 1: Search Strategy*

For the first phase of our systematic literature review, we developed a search strategy for finding papers. The search strategy was executed by an automated comprehensive search taking as input a set of academic search engines and a list of search terms. The search was performed by applying each search term to each engine incrementally until the stopping criteria were met. Table 1 lists the four search engines along the left most column with the three search terms across the top row along with the papers for each criterion and engine combination. The search algorithm was performed as follows:

1. Call to search engine with current search position and current search term.
2. Parse results and extract paper title, authors and year of publication.
3. Compare results against stopping criteria:
    - If the size of the result set is greater than or equal to 1000 then stop.
    - If the last ten results did not contain the search term phrase within the title then stop.
4. If stopping criteria not met, increment search position and go back to step one.

The result set size stopping criteria was chosen due to a technical limitation of some search engines. The stopping criteria related to the last ten titles is meant to stop after relevant results are no longer being returned by the search engine. After gathering all 12 data sets, we combined the papers into a master list, which includes only distinct papers by systematically comparing the bibliographic information for each. Out of the master list of 1,716 papers, two reviewers, namely Reviewer 1 and Reviewer 2, conducted a series of elimination rounds to narrow the list of papers and identify primary sources. Table 2 shows the total number of papers selected by each reviewer for each round and how many papers from the disjoint set for each round survived to the next round.

### *3.2. Steps 2-4: Elimination Rounds*

The elimination rounds were conducted based on reading of the title, abstract, and finally the paper itself. While each elimination stage had a unique set of criteria for elimination, the general procedure for elimination for the researchers was as follows.

- The two first authors independently classified papers as relevant, irrelevant or uncertain based on elimination criteria
- Those papers marked as relevant by both reviewers were kept and those marked irrelevant by both were thrown out.
- Papers marked as relevant or irrelevant by a single reviewer were combined with all papers marked as uncertain and discussed by both reviewers. From this discussion, papers were either thrown out or kept until the next round of the review.

#### *3.2.1. Step 2: Title Elimination*

The first round of elimination was performed by examination based on the title. Each author was tasked with deciding on elimination by answering the following questions:

- Did the title contain a reference to 'role-based access control' or 'RBAC'?
- Did the title contain a reference to 'model'?

#### *3.2.2. Step 3: Abstract Elimination*

The second round of elimination was based on reading of the abstracts of papers that survived title elimination. Researchers read each abstract and evaluated relevancy based off:

- Does the abstract mention a proposed model?
- Does the abstract mention extension of role-based access control?
- Does the abstract mention an implementation, evaluation, or domain for their model?

#### *3.2.3. Step 4: Content Elimination*

The final elimination round involved reading the entire paper into consideration and answering five questions that would serve as the basis for elimination. The data collected by answering these questions served as the basis towards answering the research questions. Each reviewer seeks to answer following questions based on the content of the paper:

1. Does this model extend the RBAC$_m$? (Exclusion)
2. Do the researchers give evidence that RBAC$_m$ needs extension? (Inclusion)
3. Was the paper and subsequent model inspired by a real world example? (Conditional Inclusion)
4. Did the researchers offer any evaluation of the proposed model? If yes, how did they do one? If no, why? (Conditional Inclusion)

Table 2: Elimination Rounds

| | | **Title** | **Abstract** | **Content** |
|---|---|---|---|---|
| Reviewer 1 | | 305 | 86 | 46 |
| Reviewer 2 | | 176 | 102 | 42 |
| | Overlap | 249 | 51 | 24 |
| | Disjoint | 232 | 137 | 64 |
| | Rejected | 208 | 116 | 59 |
| | Retained | 41 | 21 | 5 |
| | **Total** | 290 | 72 | 28 |

5. Did the authors implement their model? (Inclusion)

Question 1 was a definitive exclusion criterion as any paper that failed in the affirmative was rejected. Questions 3 and 4 were marked as conditional includes given that they were connected in making a decision. A paper that met question 3 but not 4, or met 4 but not 3 would be included since in some cases the real world examples served as research evaluations and without this conditional include the paper list size would be too small to be significant.

### *3.3. Extraction*

After selection of primary sources, the next step was to extract data from each paper that pertained to our research questions in order to look for trends. The first step was to take the individual data generated from the final elimination round and organize this information around the research questions. During the paper reading round and resulting data, the fact that the papers were logically falling into a number of categorizations became evident. Thus, the first step undertaken was to answer the question of what categories exist for extensions to $\text{RBAC}_m$ and what papers fell into what categories.

## 4. Categorization

RQ1: What categories exist within RBAC extensions?

### *4.1. Results*

During the paper reading phase, we identify categorical themes, which are central topic or context of RBAC extensions. For example, the paper "Privacy-aware role-based access control" [5] brings in the notion of privacy explicitly within the title of the paper and the name of their model. Some papers presented this direct pronouncement of theme of RBAC extensions, while others were less obvious. Thus, we developed a set of guidelines to aide in determining a set of eight categories based upon observations during the paper reading phase. In developing these guidelines, we defined each category by a single noun-phrase descriptor. The guidelines were:

- Model Name - Does the name of the model classify itself?
- Self Assessment - Do the authors of the paper directly identify a descriptor for their model within the body of the paper?

- Repetition of Phrase - Does the body of the paper present the same phrase repeatedly when discussing their model?

The previous example paper "Privacy-aware role-based access control" [5] contained "privacy" in the title and in the name of the model leading to the creation of the Privacy category and the subsequent placement of the paper under that category. By comparison, the paper "An extended RBAC model based on granular logic" [8] does not contain a direct categorization in the title or model name. However, in reading the body of the paper, we determined that this paper discussed RBAC extension based on context. We offer definitions for each of the eight observed categories based on the data extracted from the primary sources and the English definitions for each noun-phrase. Some category descriptors contain abbreviations in parenthesis that match the shortened name found within Table 4.

*4.2. Definitions*

- **Context**: The extension model integrates contextual information into the RBAC standard model. Context is defined as a user's current state and environment (e.g., location, time, system resources, network state, network security configuration, etc) which may affect the user's access privileges.

- **Constraint (Const)**: The extension model provides conditional restrictions on permissions of given roles. The constraint is either static or dynamic. For example, a doctor may modify any medical record for which the doctor is assigned as the designated primary care physician. This example describes a doctor's permission with a conditional restriction - "only when the doctor is assigned as the designated primary care physician may modify a particular medical record".

- **Organizational (Org)**: The extension model is concerned with providing mechanisms and entities that allow for RBAC across multiple organizations. Typically, users may have the same role name in different organizations, but may have different access privileges due to different local variations.

- **Privacy (Priv)**: The extension model provides entities and mechanisms to describe privacy policies, which are legal statements or documents about disclosure or management of personally identifiable information such as name, address, data of birth, etc.

- **Task**: The extension model provides task entities which are associated with permissions and roles. A task is a fundamental unit of a business activity. Different from core RBAC, in task-role-based access control model, roles are not directly associated with permissions. Roles are associated with tasks that are associated with permissions. For example, the employee role is associated with a task, for example, to write a report. Then, this task is associated with a permission.

- **Spatio-Temporal**: The extension model combines spatial (location-based) and temporal (time-based) constraints in specifying access control policies. For example, specific locations permit roles to conduct actions from 8:00 a.m. to 5:00 p.m.

- **Spatial**: The extension model provides spatial constraints, location-based constraints in specifying access control policies. For example, in organizations, locations are enforced while a specific role is permitted to conduct an action. Consider that a part-time employee works only in a specific location. In such cases, the part-time employee role should access required resources only when the user is in that location. Spatial constraints can integrate with roles, user-role assignments, or role-permission assignments.

- **Temporal (Temp)**: The extension model provides temporal constraints, time-based constraints in specifying access control policies. For example, in organizations, periodic temporal durations are enforced while a specific role is permitted to conduct an action. Consider that a part-time employee works only from 9:00 a.m. to 3:00 p.m. In such cases, the part-time employee role should be allowed to access required resources during the interval. Temporal constraints can integrate with roles, user-role assignments, or role-permission assignments.

### *4.3. Analysis and Discussion*

The 28 primary sources produced a set of eight hierarchical categories. Table 4 summarizes each primary source under their designated category and furthermore, displays the perceived hierarchy of the categories. The Spatial and Temporal categories were treated as subsets of the broader category of Spatio-Temporal since this category encompasses them individually and the Spatio-Temporal category contained more primary sources that the Spatial or Temporal categories alone.

When looking across all categories, the authors noted that each category added some new features on top of the standard RBAC model that were domain specific. These domain specific features were under the surface these features adding contextual relationships between the core user, permission and role entities. Thus, the authors concluded that all categories stemmed from the context category, of which some primary sources were already deemed direct members.

For example, in the case of the Privacy category, the models added entities such as purpose binding to represent within the model data collected for one purpose should not used for another purpose without user consent [5]. While the new entity provided by the Privacy based models is inspired by domains such as healthcare where privacy is of legal concern, the underlying mechanism that drives purpose binding is providing context around making an access control decision. The system must take into account not just a static set of permissions a user has through their roles, but also the context of the data being accessed as that data relates to privacy policy. In the spatio-temporal models, a users location and the time of day are two factors that can be taken into account when activating a role or verifying a permission. The concepts of location and time are properties of the user and place specific contexts around the role and permission entities.

We found eight categories that exist within extensions to RBAC: Constraint, Context, Organization, Privacy, Task, Spatio-Temporal, Spatial and Temporal. We further found that the other seven categories fall under the category of Context.

## 5. Motivations

RQ2: What are the motivations behind RBAC extensions?

### *5.1. Results*

We first describe motivations described in the 28 papers surveyed.

- The RBAC$_m$ needs additional contextual information and constraints to develop fine-grained policies.
- The RBAC$_m$ does not incorporate context. Therefore, RBAC belongs to the static access control model, which may not capture changes in environments.
- The RBAC$_m$ does not support various constraints such as temporal and spatial constraints to design sophisticated policies on demand.
- The RBAC$_m$ does not provide an abstraction for additional user-defined attributes (e.g., task and team) and their association with existing attributes.
- The RBAC$_m$ has limitation on delegation and role hierarchy. For example, partial inheritance in role hierarchy needs to be developed.
- The RBAC$_m$ needs to incorporate additional attributes such as task, team, purpose, organizational roles and collaborative activities over existing attributes. Moreover, new associations between attributes are introduced.
- The RBAC$_m$ needs to improve existing features such as delegation and role hierarchy to provide fine-grained access control.

We classify motivations behind RBAC extensions to these three categories:

- Addition: The extension model incorporates additional entities with regards to context/constraints and their relations with existing entities of the RBAC$_m$. Models in this category do not change the RBAC$_m$. We found 23 papers in this category.
- Modification: The extension model incorporates modification of existing features in the RBAC$_m$. For example, changes of attribute relations such as role hierarchy fall into this modification category. For models in this category, changes occur within the RBAC$_m$. We found one paper in this category.
- Combination: The extension model is a combination of RBAC$_m$ and another access control model, which uses model-specific attributes and their relations such as task, team, purpose, and organizational roles. For this combination, new associations between attributes of two models should be introduced. Models in this category require new associations between attributes of different access control models to work together. We found four papers in this category.

Table 3: Implementation types found and the count of primary sources

| Implementation Type | Paper Count |
|---|---|
| Enterprise Implementation | 4 |
| Prototype Implementation | 4 |
| No Implementation | 20 |

### *5.2. Analysis and Discussion*

The RBAC standard provides RBAC$_m$, which is an abstraction on top of authorization based on $UA$ and $PA$. Since RBAC$_m$ may not provide a fine-grained access control for a sophisticated security mechanism, a variety of extended RBAC models have been proposed over the years to meet security requirements. When administrators design a model, it is important to capture an important abstraction to help the model to be enforced in a system.

RBAC$_m$ presents four entities: roles, sessions, users, and permissions, and their relations. While RBAC$_m$ is considered fundamental in any RBAC systems, RBAC$_m$ can be extended to be applicable for emerging applications such as healthcare and mobile devices. In such environments, the RBAC standard needs additional constraints and dynamic context to support dynamic environments such as spatial and location. For this category, more than 80% of papers are classified into "Addition" category. Besides, one paper [9] describes modification of existing relationship among entities to support partial inheritance. The RBAC standard is shown to be combined with other access control models such as team-based access control and task-based access control [10, 11].

## 6. Implementations

RQ3: Do the extension models have corresponding implementations?

### *6.1. Results*

When designing and proposing a model targeted at a feature that is rooted in practical usage by real software systems, bringing the model to life is strong evidence that the proposed model can work in practice. The concept of authorization and access control is rooted in a business need. Thus, any access control model needs to be feasible in the real world not just on paper. We analyzed the primary sources to see how many proposed models actually had implementations associated with them. Then, we quantified types of implementations. Whether the implementation was for a real system, for a prototype and/or used in a production environment.

Table 3 shows the breakdown of implementations found within the primary sources. Of the 28 papers surveyed, there was a lack of implementation with 20 of the paper providing no mention of an implementation or prototype. Of the remaining eight papers that did mention an implementation, half were simply prototypes developed by the authors while the other half were claimed to be implemented within a real system.

### *6.2. Analysis and Discussion*

The RBAC standard was designed with enterprises in mind such that when practitioners implemented RBAC into their systems there would be a reasonable assurance being based off a well

thought out model. As extensions to the RBAC$_m$ come along, thought and time should be given to how features and nuances of their models may impact implementation in order to achieve the same goals as the original standard. The primary sources should a significant lack of implementation with over 70% of the models having no notion of attempting to implement them. The bare minimum, as four papers did, should be a prototype implementation of the model for review by both practitioners and researchers. Of the models that produced an implementation within the enterprise world, two were from within the medical domain and two were implemented using web application technologies.

## 7. Evaluations

RQ4: How are extensions to RBAC evaluated theoretically and in practice?

### *7.1. Results*

The 28 primary sources were examined for evidence that evaluations of the proposed model were presented by the model authors. Further, we identified a set of evaluation types found within each of the primary sources and provide below a list of the evaluation types and which primary sources provided which type. In some cases a single primary source provided multiple evaluation types.

- Time-based Performance [5], [12]
- Complexity analysis [13], [14], [15], [12]
- Comparison to standard RBAC [13], [16], [14], [17], [18]
- Mathematical modeling [19], [20], [21], [15], [22]
- Example scenarios of the model in action [9], [23], [24], [25], [13], [8], [26], [16], [18], [27], [22], [28], [10], [11]
- Experimental analysis of the model
- Case study of the model in practice [29]

Based on the diverse evaluation criteria, 12 models presented no evidence of an evaluation. Eight models presented example scenarios and how application of their model would apply and resolve the situation. Six of the models provided some form of performance or complexity analysis of their model. This included graphs of the model's time to determine authorization as the number of entities grew, and the size of the role space for the extension model compared to standard RBAC. Four models provided mathematical descriptions and analysis as a way to provide evaluation in the form of completeness. The most widely used evaluation method was providing sample scenarios with accompanying workflows of how the extension model would tackle those scenarios. Much is left to the reader to assume of these types of evaluations, as the authors do not explicitly state or show how the RBAC$_m$ is deficient in tackling said scenarios.

### *7.2. Analysis and Discussion*

When proposing an access control model, providing an evaluation of the model is a key component in establishing the validity of the model. Further, in the case of extensions to the RBAC$_m$, the model should be accompanied by validation of the model as a stand-alone access control model and in comparison to the model upon which the enhancements are being made. The results show that robust evaluations of extension models are lacking.

The primary source of validation a developer or practitioner may encounter is a qualitative discussion of real-world scenarios and how the proposed model can tackle those situations. In rare cases, five primary sources, the model authors provide some discussion of how the RBAC$_m$ is deficient in tackling the scenario. In rare cases, one paper from our results, a case study is performed to examine how the proposed model works in practice. Discussions of how an extension model handles a real-world scenario provides developers and practitioners anecdotal evidence at best for what types of situations the proposed model could handle. Further, by not providing a comparison to the RBAC$_m$, developers may be left implementing a more complex model to address their requirements when the RBAC$_m$ would have sufficed. Further, given the nature of access control models as grounded in application to enterprise implementations, case studies of a model in action provide developers with evidence that the model works as intended when applied.

When looking for an enterprise ready access control mechanism, developers must balance usability with security. Two of the primary sources examined provided time-based performance analysis of their extension model compared to the RBAC$_m$. This inclusion of time analysis provides some assurances to developers that any non-functional requirements surrounding time to compute authorizations compete or beat the RBAC$_m$. Further, four of the models provided some form of complexity of their model. This complexity analysis plays a key role in the management of the access control mechanism over the course of the models implementation lifetime. As the number of roles, users and additional entities grows, developers will need to ensure non-functional requirements are met that deal with the ability for a system administrator to effectively manage these entities.

## 8. Domains

RQ5: What domains have RBAC extensions been created for?

### *8.1. Results*

Business needs have historically driven RBAC research and development. The primary mode of evaluation for model extensions has been the presentation of business scenarios in various domains and how the model uniquely handles those particular scenarios. Thus, looking for trends in the domains used in the example scenarios might serve to illuminate a trend worth further examination into the reason for the explosion of RBAC extensions. The authors identified domains presented within the primary sources by looking for example scenarios cast within a particular domain or mention of domain requirements within the body of the paper. We found that the domains mentioned and their associated sources are:

- Medical domain [9], [23], [29], [5], [19], [20], [27], [12], [10]

- Pervasive computing environments [25], [15], [18]
- Web applications [30]
- Mobile computing [31], [16], [32], [18], [12]
- Large-scale organizations with many sub-departments [26], [33], [28]
- Enterprise, organization workflows [24], [13], [14], [11], [22]

The predominant domain for which extension models have been generated for is that of the medical domain with nine of 28 mentioning scenarios or requirements of that industry. Mobile computing and enterprise workflows were each represented by five papers claiming to be influenced by the requirements for access control within these domains. The final set of domains was pervasive computing environments and large-scale organizations with three each and web applications with one. There were four papers without any direct mention of a domain since Aich et al. [12] fall under both the medical domain and mobile computing.

### *8.2. Analysis and Discussion*

The medical domain produced the largest selection of papers when analyzing the domains influencing the proposals of extension models. Further, the authors noted that the categories associated with papers identifying the medical domain were not limited to one or two but cut across each of the eight categories except for Organization. The cross-category nature of the medical domain papers appears indicative of the complex nature of medical applications and the requirements therein. Given the growth of the research and development of medical applications over the past decade this result does not appear to be surprising. However, the RBAC standard was originally created to reduce cost and increase interoperability - two goals of current regulation around the standardization of electronic health record systems. The large number of proposed models, and the cross-category result stand in direct opposition of the goals of both the RBAC standard and current regulations.

The RBAC standard has been re-enforced by the economic impact that standardization has had on enterprises needing to apply access control. The inclusion of extension models targeted at the enterprise workflow domain is indicative of the expansion of requirements for enterprises. Developers and researchers should take care when looking at extension models designed to addresses the newer requirements of enterprise workflows in order to achieve the same economic implementation and maintainability benefits the RBAC$_m$ presents.

Mobile computing has seen a dramatic increase in the number of available devices, operating systems and applications since 1997 when the first smart phone was introduced. The domain analysis results produced five papers that targeted extensions that are designed to address the requirements of mobile computing. For a domain that has roots in personal and enterprise computing, protecting the data of both through access controls is paramount given their ubiquity.

## 9. Generalizations

RQ6: What commonalities or generalizations exist across all categories?

### 9.1. Results

The RBAC standard is used in various aspects of computer systems. In order to reduce efforts for modeling access control used in various applications, researchers often focus on developing generalized core concepts of access control. We found that propositional logic is used to describe access control model across all categorizations. Propositional logic is concerned with propositions and their logical relationships. In propositional logic, simple (i.e., atomic) or compound condition at given context is evaluated to true or false based on specified rules and access control logic. Researchers are concerned to extend limited set of propositions specific to core RBAC to meet real-world scenarios such as dynamic constraints, temporal, or spatial constraints. However, semantic meanings of such propositions are various based on researchers' intention.

### 9.2. Analysis and Discussion

Since NIST proposed RBAC standard using propositional logic, researchers describe extended RBAC models using propositional logic. Given context, propositional logic is used to evaluate true or false based on specified rules and access control logic. As access control is typically evaluated to either true (Permit) or false (Deny) based on predicates, propositional logic is sufficient to describe key ideas and definitions. We found that 28 papers of RBAC extensions use propositional logic to describe its extended model. However, propositional logic has limitations. While this logic is simple, this logic does not support for reasoning about RBAC, which helps reduce the administrative complexity of associations such as user- role associations. An alternative logic to describe RBAC is first-order logic. This logic is sufficient to describe RBAC and extended RBAC. Moreover, this logic supports for concise and elegant formulation of the RBAC$_m$ and its relation. First-Order logic is expressive enough to concisely represent access control systems. First-Order logic uses relations, variables, and quantifiers.

## 10. Discussion

The research questions we identified, presented results for and analyzed provide specific views of portions of the extension model landscape. By stepping back, and looking across all the research questions we can arrive at cross-cutting concerns and identify areas that may benefit from future work. We break the discussion into two issues: recommendations for the enhancement of the evaluation of extensions to role-based access control models and guidelines for developers needing to choose an extension model to meet their requirements.

### 10.1. Metrics for the Evaluation of RBAC Extensions

A small percentage of primary sources presented substantial evaluations or implementations of their model in practice. This in turn leaves researchers and practitioners little data from which to compare one model to the next. And given the wide swath of popular and critical domains being targeted by RBAC extension models, we propose a set of metrics by which RBAC extension models should be evaluated. Hu and Scarfone provide metrics for access control system evaluation or a security system in general [34]. They collect metrics, which can be applicable for not only access control models, but also requirements, system implementation, and extended application such as verification and testing. From these metrics, and guided by the data surrounding how

extension models are currently evaluated, we select metrics that can be applicable for evaluation of extended RBAC models as follows:

- **Ease of Privilege Assignments**: measures number of steps for assigning, changing, or deleting a privilege for users and roles when a user or administrator manages RBAC. If a model requires more steps for such a given task, there is a potential increase in human or system error due to increased complexity. While extended RBAC may have similar features to that of the RBAC$_m$, we measure worst cases after extended, and the gap between an extension model and the RBAC$_m$.

- **Syntactic and semantic support for specifying AC rules**: checks whether privileges and constraints can be specified using complex logic expressions such as AND, OR, $\leq$ when specifying constraints. This support helps specify complex and flexible access controls. For example, given a spatial RBAC model, one may measure whether spatial constraints can be specified using various logic expressions.

- **Delegation of administrative capabilities**: measures how easy and secure an access control model supports for privilege delegation of administrators to another user.

- **Capable of combining different policy models**: measures whether an access model has a feature to combine more than two policies.

- **Capable of conflict resolution**: checks whether an access model has a feature to resolve conflicts when two rules in a policy may be applicable for the same request, but evaluate different decisions: One grants permission and the other one denies permission.

- **Least privilege principle support**: checks whether an access model supports for least privilege principle where a subject can access least set of information and resources that are necessary for its legitimate functioning. Least privilege principle support can reduce damage when malicious users gain legitimate privileges.

- **Separation of Duty (SoD)**: checks whether an access model supports for static SoD and/or dynamic SoD. Note that NIST Standard supports both static SoD and dynamic SoD described in Section 2.

We recommend these metrics be incorporated in any extension models presented hence forth. A future work could include providing metric measurements for the primary sources presented here. A developer could attempt to apply these metrics to an extension model not already evaluated when attempting to choose a model. These metrics focus on comparing specific features support of access control models.

### *10.2. Adoption of Extensions to RBAC*

In order to adopt a model in practice, software developers should implement the model in real system environments based on intended use of the models. Typically, a model provides abstract formal representation of extended RBAC and its operation. To configure intended use (i.e., which

subjects can access which resources), we define an access control policy, which specifies high-level rules such as which subjects can access to which resources. Given an access control model and policy, security mechanism provides low-level implementation in a system, which controls access of roles.

A formal representation of access control models is a critical step of designing high-level abstraction where which entities are used and how these entities are operated. One of the objectives of a formal representation is to help ensure the correct behaviors through formal verification. Given the formal representation of access control models, software developers may prove of properties (e.g., safety, consistency and completeness) and check whether these properties are satisfied.

While the concept of the RBAC$_m$ is clear and can be applicable to any system, it is challenging to implement the security mechanism of RBAC because the RBAC$_m$ is can be interpreted in more than one interpretation due to its complexity. In order to bridge gap between RBAC standard and its implementation, NIST published RBAC Implementation and Interoperability standard (RIIS/ ANSI INCITS-459) in 2011. This standard specifies how to implement RBAC system, which is consistent with NIST RBAC standard. Moreover, this standard describes interoperability specification where one RBAC implementation can be translated to another one.

This standard does not provide a specific guideline for implementation of various extended RBAC models. Moreover, when software developers implemented extended RBAC models, they require clear security requirements of extended role-based access control, which identifies security objectives, intended environments, and assumed threats. Moreover, software developers understand how this extended RBAC overlaps with other access control models when more than two access control models are integrated into a single system. A survey [1] shows that over 50% of users at organizations with more than 500 employees are given some of their permissions to access resources based on RBAC. To adopt RBAC into a system, organizations often use hybrid approaches, which combine RBAC and access control lists since specific user types, systems, and workflow may not be effective to manage access based on roles. Therefore, when RBAC is extended and combined with other access control models, software developers often require to provide a formal representation, which can support of proof of the model. Then, software developers implement extended models based on NIST standard of RBAC implementation. Especially, software developers should understand intended use of extended access control models to meet a new system requirement. For example, for a spatial RBAC model, software developers incorporate additional spatial constraints, which can be either static or dynamic, in practice into the RBAC model.

## 11. Conclusion

The RBAC extensions were revealed to fall into a number of categorizations with Organization, Privacy, Resource, Task, Spatio-Temporal, Spatial, and Temporal falling under the general category of context. The categories each had properties specific to their implementation, but were seen to generalize to being specialized instances of context tailored to the entities or actions the categories covered. A number of domains were identified as being the motivations behind needing extensions to the core RBAC model. The domains, such as healthcare, presenting new challenges the previous models were not required to design for. Our literature review showed that the state

of RBAC model evaluation needs focused from the research community given most model evaluations seen within the papers were based on hypothetical situations with little to no case studies or implementations in practice.

## References


[1] A. C. O'Connor, R. J. Loomis, 2010 economic analysis of role-based access control, RTI International report for NIST.

[2] D. Ferraiolo, D. Kuhn, Role based access control, in: 15th National Computer Security Conference, Oct 13-16, 1992, pp. 554–563.

[3] R. Sandhu, E. Coyne, H. Feinstein, C. Youman, Role-based access control models, Computer 29 (2) (1996) 38–47.

[4] D. R. Kuhn, E. J. Coyne, T. R. Weil, Adding attributes to role-based access control, Computer 43 (6) (2010) 79–81.

[5] Q. Ni, E. Bertino, J. Lobo, C. Brodie, C. Karat, J. Karat, A. Trombeta, Privacy-aware role-based access control, ACM Transactions on Information and System Security (TISSEC) 13 (3) (2010) 24.

[6] D. F. Ferraiolo, R. S. Sandhu, S. I. Gavrila, D. R. Kuhn, R. Chandramouli, Proposed NIST standard for role-based access control, ACM Transactions on Information and System Security 4 (3) (2001) 224–274.

[7] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering.

[8] H. Jian-min, L. Xi-yu, Y. Hui-qun, T. Jun, An extended rbac model based on granular logic, in: Granular Computing, 2008. GrC 2008. IEEE International Conference on, IEEE, 2008, pp. 261–264.

[9] M. Alam, M. Hafner, R. Breu, A constraint based role based access control in the sectet a model-driven approach, in: Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services, PST '06, 2006, pp. 13:1–13:13.

[10] W. Zhou, C. Meinel, Team and task based rbac access control model, in: Network Operations and Management Symposium, 2007. LANOMS 2007. Latin American, IEEE, 2007, pp. 84–94.

[11] S. Oh, S. Park, Task–role-based access control model, Information Systems 28 (6) (2003) 533–562.

[12] S. Aich, S. Mondal, S. Sural, A. K. Majumdar, Transactions on computational science iv, 2009, Ch. Role Based Access Control with Spatiotemporal Context for Mobile Applications, pp. 177–199.

[13] Y. Bao, J. Song, D. Wang, D. Shen, G. Yu, A role and context based access control model with uml, in: Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for, 2008, pp. 1175 –1180.

[14] Z. Zhang, X. Zhang, R. Sandhu, Robac: Scalable role and organization based access control models, in: Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006. International Conference on, 2006, pp. 1 –9.

[15] L. Chen, J. Crampton, On spatio-temporal constraints and inheritance in role-based access control, in: ASIACCS: ACM Symposium on InformAtion, Computer and Communications Security, 2008, pp. 205–216.

[16] D. Zou, L. He, H. Jin, X. Chen, Crbac: Imposing multi-grained constraints on the rbac model in the multi-application environment, Journal of Network and Computer Applications 32 (2) (2009) 402–411.

[17] Y. Zhao, Y. Zhao, H. Lu, A flexible role-and resource-based access control model, in: Computing, Communication, Control, and Management, 2008. CCCM'08. ISECS International Colloquium on, Vol. 2, IEEE, 2008, pp. 75–79.

[18] I. Ray, M. Toahchoodee, A spatio-temporal role-based access control model, in: Proceedings of the 21st annual IFIP WG 11.3 working conference on Data and applications security, 2007, pp. 211–226.

[19] M. Damiani, E. Bertino, B. Catania, P. Perlasca, Geo-rbac: A spatially aware rbac, ACM Transactions on Information and System Security (TISSEC) 10 (1) (2007) 2.

[20] F. Hansen, V. Oleshchuk, Spatial role-based access control model for wireless networks, in: Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th, Vol. 3, IEEE, 2003, pp. 2093–2097.

[21] S. Aich, S. Sural, A. K. Majumdar, Starbac: spatiotemporal role based access control, in: Proceedings of the 2007 OTM confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part II, OTM'07, 2007, pp. 1567–1582.

[22] J. Joshi, E. Bertino, U. Latif, A. Ghafoor, A generalized temporal role-based access control model, Knowledge and Data Engineering, IEEE Transactions on 17 (1) (2005) 4 – 23.
[23] S. K. Tzelepi, D. K. Koukopoulos, G. Pangalos, A flexible content and context-based access control model for multimedia medical image database systems, in: Proceedings of the 2001 workshop on Multimedia and security: new challenges, Sec '01, 2001, pp. 52–55.
[24] D. G. Cholewka, R. A. Botha, J. H. P. Eloff, A context-sensitive access control model and prototype implementation, in: In: Information Security for Global Information Infrastructures: IFIP TC 11 Sixteenth Annual Working Conference on Information Security, Kluwer Academic Publishers, 2000, pp. 341–350.
[25] X. Huang, H. Wang, Z. Chen, J. Lin, A context, rule and role-based access control model in enterprise pervasive computing environment, in: Pervasive Computing and Applications, 2006 1st International Symposium on, 2006, pp. 497 –502.
[26] W. Yamazaki, H. Hiraishi, F. Mizoguchi, Designing an Agent-Based RBAC system for dynamic security policy, in: Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE '04, 2004, pp. 199–204.
[27] E. B. Arjmand Samuel, Arif Ghafoor, A framework for specification and verification of generalized spatio-temporal role based access control model, Tech. Rep. CERIAS Tech Report 2007-08.
[28] L. Yao, X. Kong, Z. Xu, A task-role based access control model with multi-constraints, in: Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on, Vol. 1, IEEE, 2008, pp. 137–143.
[29] G. Motta, S. Furuie, A contextual role-based access control authorization model for electronic patient record, Information Technology in Biomedicine, IEEE Transactions on 7 (3) (2003) 202 –207.
[30] A. Masoumzadeh, J. Joshi, Purbac: Purpose-aware role-based access control, On the Move to Meaningful Internet Systems: OTM 2008 (2008) 1104–1121.
[31] N. Thein, et al., Leveraging access control mechanism of android smartphone using context-related role-based access control model, in: Networked Computing and Advanced Information Management (NCM), 2011 7th International Conference on, IEEE, 2011, pp. 54–61.
[32] S. M. Chandran, J. B. D. Joshi, Lot-rbac: a location and time-based RBAC model, in: Proceedings of the 6th international conference on Web Information Systems Engineering, WISE'05, 2005, pp. 361–375.
[33] H. Jian-min, L. Xi-yu, Y. Hui-qun, T. Jun, An extended RBAC model based on granular logic, in: Granular Computing, 2008. GrC 2008. IEEE International Conference on, 2008, pp. 261 –264.
[34] V. C. Hu, K. Scarfone, Nistir 7874: Guidelines for access control system evaluation metrics, National Institute of Standards and Technology.
[35] S. Haibo, H. Fan, A context-aware role-based access control model for web services, in: e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on, IEEE, 2005, pp. 220–223.

## 12. Appendix

Table 4: Primary sources grouped by categorization

| | | | |
|---|---|---|---|
| Context | A flexible content and context-based access control model for multimedia medical image database systems, 2001 [23] | | |
| | A Context-Aware Role-Based Access Control Model for Web Services, 2005 [35] | | |
| | A context-sensitive access control model and prototype implementation, 2000 [24] | | |
| | A Context, Rule and Role-Based Access Control Model In Enterprise Pervasive Computing Environment, 2006 [25] | | |
| | A contextual role-based access control authorization model for electronic patient record Motta, 2003 [29] | | |
| | A Role and Context Based Access Control Model with UML, 2008 [13] | | |
| | An extended RBAC model based on granular logic, 2008 [8] | | |
| | Designing an agent-based RBAC system for dynamic security policy, 2004 [26] | | |
| | An extended RBAC model based on granular logic, 2008 [33] | | |
| | Leveraging Access Control Mechanism of Android Smartphone Using Context-Related Role-Based Access Control Model, 2011 [31] | | |
| | CRBAC: Imposing multi-grained constraints on the RBAC model in the multi-application environment, 2009 [16] | | |
| | Const. | A constraint based role based access control in the SECTET a model-driven approach, 2006 [9] | |
| | Org. | ROBAC: Scalable Role and Organization Based Access Control Models, 2006 [14] | |
| | Priv. | Privacy-aware role-based access control, 2007 [5] | |
| | | PuRBAC: Purpose-Aware Role-Based Access Control, 2008 [30] | |
| | Task | A Task-Role Based Access Control Model with Multi-Constraints, 2008 [28] | |
| | | Team and Task Based RBAC Access Control Model, 2009 [10] | |
| | | Task-role-based access control model, 2003 [11] | |
| | Spatio-Temporal | STARBAC: Spatiotemporal role based access control, 2007 [21] | |
| | | On spatio-temporal constraints and inheritance in role-based access control, 2008 [15] | |
| | | A framework for specification and verification of generalized spatio-temporal role based access control model, 2007 [27] | |
| | | LoT-RBAC: A Location and Time-Based RBAC Model, 2005 [32] | |
| | | A Spatio-temporal Role-Based Access Control Model, 2007 [18] | |
| | | Role Based Access Control with Spatiotemporal Context for Mobile Applications, 2009 [12] | |
| | | Spatial | GEO-RBAC: a spatially aware RBAC, 2005 [19] |
| | | | LRBAC: A location-aware role-based access control model, 2006 [18] |
| | | | Spatial role-based access control model for wireless networks, 2003 [20] |
| | | Temp. | A generalized temporal role-based access control model, 2005 [22] |