

# “Let Me Tell You What I Want” – Security Policy Elicitation Through Computational Narration

Ronda R. Henning  
Harris Corporation  
rhenning@harris.com

## Abstract

A system security policy is subject to considerable interpretation. What to the end user may be a perfectly reasonable access control policy may be impossible to architect into an enforceable policy implementation. The earlier such policy disconnect can be found, the less severe the impact on the system design, cost, and schedule. This paper discusses the use of computational narrative, or computer-assisted storytelling, as a method for eliciting the access control policy associated with a given information system. Similarities in the structure between computational narration and access control models are presented, as are attempts to apply computational narration in similar domains. Finally, a research project is proposed to determine the feasibility of computational narration as an access control modeling technique.

## Introduction

The phrase *security policy* has been used to address the protection mechanisms employed to protect an organization’s assets from potential misuse. In reality, a security policy is composed of several sub-policies: accountability, authentication, contingency, and access control. The access control policy defines how system users interact with the data stored within the system. In conventional access control models, access control is defined as a triple consisting of the <subject, object, privileges> associated with a given data container, for example, a file or a row in a database. In the early days of computing, much discussion surrounded how access control models should be represented in computer systems. To address the workings of classified information processing, (Schell) developed the notion of multilevel mode of operations, stating:

In multilevel mode, the computer must internally distinguish multiple levels of information sensitivity and user authorization. Internal Controls of hardware and programs must assure that each user has access to only authorized information (p.20).

Schell further stated:

The security kernel design is derived directly from a precise specification (i.e. mathematical model) of its functions (like a cryptographic algorithm). This mathematical model is a precise formulation of access rules based on user attributes (clearance, need-to-know) and information attributes (classification) (p.21).

Multilevel mode, unfortunately, did not reflect the security policies of most civil government and commercial organizations. The U.S. Government has a uniform classification hierarchy for information, (Top Secret, Secret, Confidential, and Unclassified). Commercial organizations do not maintain a uniform classification hierarchy, but instead implement access controls based upon a user's role in the organization. For example: a user may have a functional role (system administrator), an organizational role (manger, IT department), and an administrative role (time card approver). The concept of Role-Based Access Control (RBAC) was introduced by (Ferraiolo; Schell) and (R. C. Sandhu, E.J; Feinstein, H.L.; Youman, C.E.) to more accurately model the workings of a commercial enterprise. In RBAC, access control is based on a four-element set (user, group, object, privileges). A user may belong to many groups, each with a different privilege set. In the worst case model, every user has their own group, and there are as many groups to administer as there are users. In (Ferraiolo) a

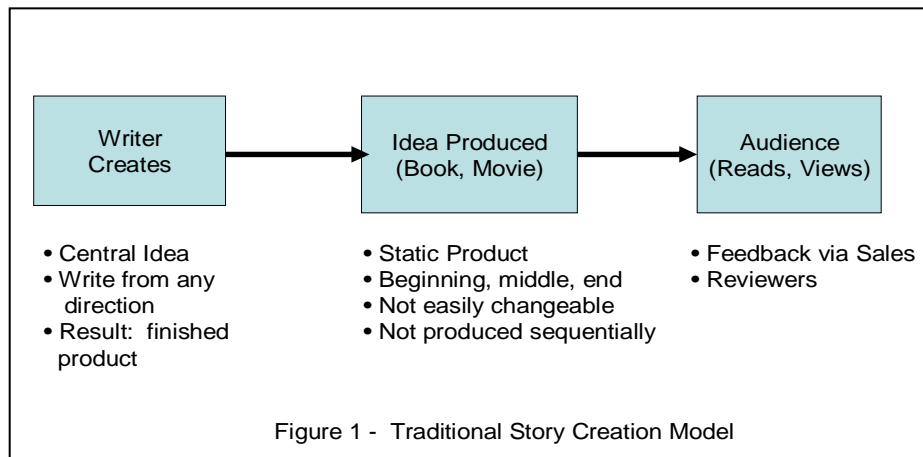
role is defined and centrally administered within an organization. RBAC has been accepted as a useful modeling tool that reflects both how organizations function and how information access is implemented in most commercial applications.

In today's information system environment, a major system development issue is the translation of the access control policy from the user's intentions to the physical implementation of the system. An access control policy that makes total sense to the user may not prove feasible when it is time to allocate system functionality for design. For example, a user's access control policy may be to share information with only members of the same corporation, except when a teaming agreement is in place for a given proposal. If an enterprise does not have its disk space segregated to allow granularity of access controls, and a network access control policy with appropriate virtual private networks and/or firewalls in place, this policy would not be enforceable. Proposal teammates might have to work in stand-alone mode, air gapped from electronic resources that could facilitate their tasks.

Moving from the world of user defined security policy to the world of an enforceable security policy is a difficult task of interpretation. Expecting a consumer organization to understand the security world of <subject, object, privileges> and be able to articulate that policy unambiguously is unrealistic.

In recent years, there has been an emerging application of natural language processing, that of computational narration. A story, in the traditional sense, is "a factual or fictional account of an event or series of events." (Microsoft, 2003). Brooks (1996), makes the observation that stories tend to be written in a linear fashion, and are perceived by the audience within the context of their cultural experience. Stories tend to be told

sequentially, with a beginning, middle, and end state. However, they are not usually created sequentially. Authors start with an idea, and may begin in the middle or at the end of the story. The finished product is refined over several drafts; revised at the request of editors, directors, or producers; and eventually goes to press, where the intended audience votes with their wallets on the author's success or failure. Figure 1 illustrates the sequential nature of story production.



The notion of computational narrative explores the customization of the storytelling experience for each audience member. In a computational narrative model, the audience actively participates in the navigation of the story. For example, a game such as Adventure or Dungeon and Dragons takes different execution branches depending on the user's decisions. With a computational narrative model, the story can be tailored to reflect each participant's perceptions.

The problem to be addressed in this research is whether or not computational narrative technologies can be applied to access control policy specification. If guided narration can be employed to assist in the definition of an access control policy, it may be possible to build an access control policy that more closely conforms to the user's intent than the current generation of relatively inflexible security policy models. Alternatively,

it may be possible to better visualize policy interactions, which would result in earlier identification of flawed policy decisions.

The remainder of this paper is structured as follows. In Section 2, prior research in security policies and computational narrative is presented. In Section 3 the significance of this research is discussed. In Section 4, a methodology to determine the feasibility of applying computational narration to access control policy is presented.

## PRIOR RESEARCH IN SECURITY MODELING AND COMPUTATIONAL NARRATION

### *Access Control Models*

Environmental constraints, such as access control specifications, exist at all levels of abstraction. The issue is how to effectively express these constraints such that the semantics can be captured and the information formalized. In (D. E. Bell) 4 levels of abstraction are associated with any given security policy:

1. an organizational abstraction level, written as a narrative, for people to read;
2. a conceptual abstraction level, discussing an organizational policy at the concept level;
3. an abstract level, describing the design and tracing the conceptual requirements to functional components; and
4. an implementation level, describing the design as developed.

Bell further models security policy as a series of requests, decisions, and state-transitions decisions that capture the intermediate actions from the initial request to the final decision. In essence, the context associated with access control decision is modeled in these state transitions. Within this model, computational narration address layers 1 and 2 of organizational policy. It is comprehensible to the policy creator, and can be

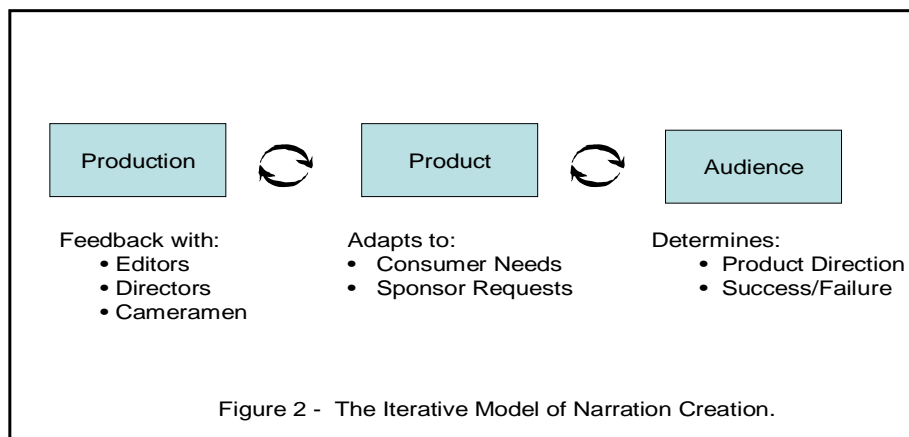
translated into the abstract layer to apply formal methods to determine policy soundness, and to the implementation layer to reflect decomposition of the design into its respective functional policy enforcement components.

### *The Foundations of Computational Narrative*

One of the early experiments in computational narrative was conducted by Kevin M. Brooks at the MIT Media Lab (Brooks, 1996). Brooks decomposed a story into 3 atomic components:

1. events,
2. people, or characters, and
3. things.<sup>i</sup>

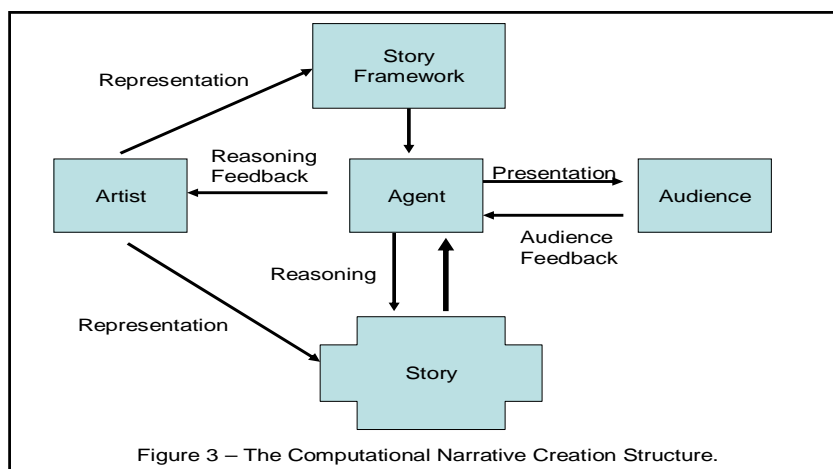
These three components are built into a narrative, which describes the organization of information. A narration explains how the narrative is expressed to the intended audience. For example, a movie may start as a narrative created by the screen writer. As it moves through production, various other experts such as costumers, dialogue coaches, and the director may alter the narrative over several iterations. The resulting end product is the narration. Figure 2 illustrates the iterative nature of narration creation.



Brooks believed that a story could connect the creator with the audience through a computational model. In Brooks' model, a narration was created through four different processes:<sup>ii</sup>

1. *the representation process* – which defined the components of the story.
2. *the presentation process* - determining how the components are revealed to the audience.
3. *the reasoning process* – which applies logical inferences about the components based on the representation.
4. *the reasoning engine* – which coordinates the processes of reasoning, representation, and presentation, and provides the results to the audience.

Through the use of behavior-based artificial intelligence, the reasoning engine in Brooks' model was capable of adapting either to the audience's response or to the creator's manipulation of the elements, based on the inference engine's recalculation of the story line. Figure 3 illustrates the computational narrative model of story creation (Brooks, 1996).



Brooks further defined three components of a computational narrative:

1. The representational environment – which reasons about the world presented in a given story.
2. The structural environment, or story framework – which provides the basic story events, such as characters, conflicts, resolution, diversions, and endings.
3. The presentation environment – which presents the results of the interpretation of the story framework and the representation to the audience, and delivers feedback to the artist for subsequent structure and story manipulation.

From the preliminary perspective provided by Brooks, research continued into how to capture collaboration among authors and tell stories from alternative viewpoints.

Mazalek and Davenport have explored the use of interaction platforms to define the structural environment and help the audience develop a frame of reference (Mazalek).

With their colleague Ishii, Mazalek and Davenport developed a Tangible Viewpoints system to allow collaborative authors to choose a story's direction through the manipulation of pawn-like tokens on a game screen. With this model, the data is represented in 3-dimensions, and the collaborators can maintain a frame of reference for each character of interest (Mazalek, 2002). Case study experiments of how the Tangible Viewpoints model works in actual collaborations are included in their work.

Refinements to the story framework have been proposed by Zagalo and Szilas. In (Zagalo), the concept of emotion as a feedback mechanism is introduced, and a notation for detection, categorization, and intensity is incorporated in the feedback provided by the story agents. Szilas looks at action as a narrative structure element that augments the story framework as a refinement mechanism based on audience interaction (Szilas).

### *Context-based Access Control Models*

Ubiquitous computing paradigms bring with them a new collection of access control policy issues. These issues have been characterized as context based access control models. Table 1 summarizes the concepts of contextual access control models.

Table 1 – Conceptual Summary of Context Based Access Controls

Author	Type of context	Access Decisions	Granularity
Hengartner (Hengartner)	Location based	Is a given users location shared by devices which may have geolocating capabilities.	Some users, no users, all users
Lei (Lei)	Constrained awareness	Transmission method and extent of situational awareness	Bandwidth or device limited contingent on device capabilities
Strembeck (Strembeck)	Conditional	Static constraints predefined, or dynamic constraints at runtime	Per object, or per access mode associated with an object.
Covington (Covington)	Environmental	Application adaptation based on environmental considerations	Per object, per action associated with a given object type.
Sandhu(R. <u>Sandhu</u> )	Usage Controlled	Access to an object defined as a dynamic function	Per object, per access, per session, or across sessions

## IS AN ACCESS CONTROL MODEL REALLY A STORY?

As access control models have become more robust, it has become increasingly important to involve the system user earlier in model formulation. Accurate interpretation of the intent of a user's security requirements must be completed as early in the requirements specification phase as possible. As the security countermeasures are derived from the policy enforced, a flawed policy will result in a flawed design. For example, a policy that states "only authorized users may access network resources" embodies both an authorization mechanism and an access control policy that uses the success or failure of the authorization attempt to make further access control decisions. In Bell's (1994) parlance, existing access control models have focused at the implementation and abstract levels of policy. That is, they are used to functionally decompose the access control policy into a system architecture. Organizational policy directives can be considered a conceptual abstraction. What has been lacking is a "plain English" organizational abstraction that succinctly expresses the end user's vision of access control semantics for a system.

Branting (Branting) discusses the use of narrative cases to control case-based reasoning in complex domains. In this model, event transitions are formulated into a narrative grammar to allow simplification of both problem formulation and system pattern matching techniques without sacrificing expressive capabilities. Waraich (Waraich) presents an interactive learning environment that uses computational narration to motivate computer engineering students as they learn binary arithmetic and logic gate generation. In both cases, relatively complex concepts are reduced to computer generated narrations which assist the end user with improved conceptual comprehension. In the early history of security technologies, security models reflected either mandatory access

controls (MAC) or discretionary access controls (DAC) as a security mechanism. More flexible models such as RBAC were nonexistent in the formal sense. The last 10 years have brought more granular models of access control interactions. These models have become more descriptive, and, subsequently, have additional mathematical complexity required to prove their logical soundness. As such, policy modeling has become a specialty discipline, and the entity that defines the properties of system behavior may not have the knowledge required to comprehend the formal model. If the model cannot be understood by the system's consumer, the probability of rework increases as the possibility of misinterpretation increases. For example, a logically sound model may not correctly reflect the intention of the policy definer.

The similarities between the components of a computational narrative (subject, object, action) and the components of a security policy (subject, object, privilege), may make computational narration a plausible candidate technology to assist in access control policy refinement.

#### *Definition of Context*

The dictionary definition of the word "context" is the "circumstances or events that form the environment within which something exists or takes place (Press). Describing the general context of an application would be an infinite problem, as there are always new observations or attributes to incorporate into the context. (Strembeck) states that "every goal and obstacle can be used to define a context condition and can map to a concrete access control service." (p.400) The computing landscape has matured to the point where basic security mechanisms exist in most system architectures today.

With the maturation of the security mechanisms, it becomes more feasible to define a structured characterization of a system security environment using policy reconciliation.

## RESEARCH APPROACH

Applying computational narration to access control policies, requires the formulation of an ontology, An ontology, or dictionary of concepts used in the access control domain (Aubrecht). Through the characterization of CBAC policies, it should be possible to refine an ontology of access control types. For example, privileges, classification labels, and handling caveats all constrain access. The issue is how to structure an ontology for access management that would support a security oriented computational narration. In (Rees) a policy is defined as having the following attributes:

- High-level – a policy should be a top level requirement levied upon the system.
- Technology neutral – implementation of the policy should not be contingent upon any given technology. For example, file system labeling schemes vs. SAML data tags
- Risk defining – if the policy is violated or bypassed, the following bad things might happen. For example, the end of life as we know it.
- Direction establishing – data access is granted when a set of conditions are met; data access is denied when a set of conditions are not met.

Access control policies that are based on the environmental context have different definitions of the contextual information required. A representative cross-section of CBAC and RBAC Policy models(R. Sandhu; R. F. Sandhu, D.;Kuhn, D.; D. E. Bell, and

LaPadula, Leonard; Bertino; Clark) was examined to determine general policy features that intersect several access control models. The results are summarized in Table 2.

Table 2 – Common Elements of Security Policy Models

<b>Element</b>	<b>General Definition</b>
Subject	The actor, person, or process requesting an action
Object	The item to be acted upon
Location of Subject	The physical location of the subject requesting the action.
Location of Object	The physical location of the object being acted upon
Time	The window of opportunity for object access by a subject.
Action Requested	What a subject would like to do with an object. These range from the traditional read, write, execute access privileges to publish/subscribe, copy, etc. Actions vary with the model in question
Constraints on action	An action may be granted in some circumstances, and denied in others. For example, the location of a subject may not support the classification associated with the object.

(Bell, 1994) characterizes security policy models as a framework of request, transition-decisions, decision. By applying such a model, it is possible to generate a tree structure for policy that can be defined as illustrated in Figure 4.

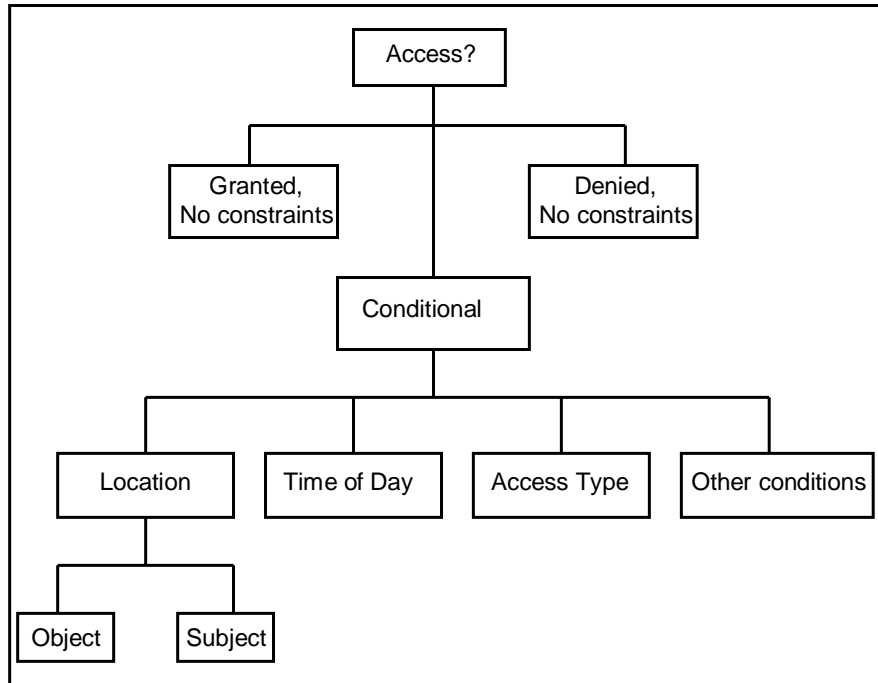


Figure 4 – Tree structure for policy decisions.

In the course of research on information assurance metrics, eight conditions of access can characterize a security policy (Fox, Henning, Tallet 2001). These eight base measures and their meta-language expressions are:

1. Physical Access to system (Defeat Physical Security)
2. External Network Access (Defeat Firewall)
3. Access to a machine (Defeat I+A (Identification and Authentication))
4. Access to Segment (Defeat Network I+A (Identification and Authentication))
5. Access to Network (Defeat Routing)
6. Access to Sensitive Data (Defeat Access Control)
7. Access to Critical Applications (Defeat Access Control)
8. Access to Logs (Defeat Auditing)

All of these measures are related to *access* to the system. Most policy compliance measurements are, in reality, probability calculations of the probability that a given user could gain access to some system element whose access should have been denied by the defined system security policy.

### *Defining the Narrative Structure*

With the information obtained in the first part of the study, a preliminary computational narration model can be created. The eight base measures can be used to define a story structure. This structure, in turn, can be used to guide the system security policy definition process. The goal of this initial model is to prove the generation of an access control computational narrative is feasible. Once the feasibility of such a model is illustrated, refinement of the model can be accomplished. The narrative structure will be created in a manner similar to that applied to the creation of vulnerability visualization environment, namely:

- defining the knowledge representation scheme for subject, objects, and access control granularity,
- defining the relationships for narration, and
- determining the extent of human computer interaction.

With the tree structure defined as in Figure 4, it is possible to define a simple story structure such as:

The President gets access to anything, anytime, anywhere.

In this case, the subject is the President, his access is to any object, regardless of time of day, or location.

An alternative story structure may be:

General XYZ only gets Army information, if he is the officer on duty, his regular shift is days, and he is in his office.

The computational complexity of the story can be bounded by the number of decisions allowed. For example, any constraint that prohibits access would be sufficient to terminate decision tree processing.

The policy implementation is irrelevant in these situations. What is conveyed is how access may be constrained, based upon the user's situation. Please note, the model is not formal in its presentation to the end user. The decision logic applied to traverse the tree is transparent to the end user. What the user sees is a story in English text that explains the policy.

The internal logic required to generate such a tree, while transparent to the end user, is of utmost importance to the formal policy modeling community. The tree defines the state transitions associated with a policy, which, in turn, can be subject to formal theorem proving technologies. The end user gets an understandable security policy, the modeling community gets a structured set of decisions that can be formally modeled, and, in theory, the resulting model reflects a better synthesis of user requirements and formalism.

### *Benefits of This Approach*

As systems become increasingly complex, it will become imperative to define an access model as early in the development and requirements solicitation process as possible. Representation of policy and policy constraints in a language tree captures the structure of the policy decision process, and also represents it to the customer early in the development process. It is much simpler to design a system to adapt to constraints if the

constraints are known at the requirements definition phase as opposed to when the finished product is demonstrated to the end user. Costly rework and replacement of security mechanisms can be readily avoided, and the consumer obtains a system with a more understandable security implementation.

The logical structure of the narrative tree also lends itself to translation to formal specification languages and subsequent propagation throughout the system design lifecycle. Changes to the narrative constraints can be captured and applied to the system design, providing insight into potential implementation issues and policy conflicts while they can be resolved with minimal cost and schedule impact.

The use of computational narration is not meant to replace formal security policy modeling, but, instead, to augment formalism such that the end user can understand the implications of access control rules. A sound formal model, be it role or contextual based, is still required. What computation narrative provides is a technique to make formal policy models more understandable to the system designer. Few software engineers understand the intricacies of theorem proving, but they do understand the rule that device X cannot access object Y. Computational narration provides for the representation of data at the organizational and conceptual level of policy instantiations, providing the necessary translation activities required to support the system development process.

## FUTURE RESEARCH

The next phase of this activity will translate the base access measures into a story hierarchy. The hierarchy will be used to define a more robust story tree, which will be applied to security policy formulation activities. The objective will be to determine if the story structure provides a more natural policy specification environment than formalism does for complex system architectures. If so, computational narrative may prove to be a valued technique for system requirements specification.

## CONCLUSION

This paper outlines ongoing research in access control policies and requirements specification. Security policy enforcement often does not reflect the intentions of the end user – either the policy is much too restrictive, or inflexible. Computational narration may provide a technique to allow early policy validation with a system's user community prior to extensive architecture design and development activities.

## References

- Aubrecht, Petr and Lubos, Kral. "Ontology Formalism Transformation." 15th International Workshop on Database and Expert Systems Applications (DEXA): IEEE Computer Society, 2004.
- Bell, D. Elliott. "Modeling the "Multipolicy Machine"." New Security Paradigms Workshop. Little Compton, RI, US: ACM, 1994. pp. 2-9.
- Bell, D. Elliott, and LaPadula, Leonard. "Secure Computer Systems: Unified Exposition and Multics Interpretation." Technical Report AD-A023 588 (1976).
- Bertino, Elisa; Catania, Barbara; Ferrari, Elena; Perlasca, Paolo. "A Logical Framework for Reasoning About Access Control Models." Trans. ACM. SACMAT'01. Chantilly, VA, USA: ACM Press, 2001. pp. 41- 52.
- Branting, L. Karl. "A Generative Model of Narrative Cases." ICAIL-99. Oslo, Norway: ACM Press, 1999.
- Clark, D. and Wilson, D. "A Comparison of Commercial and Military Computer Security Policies." IEEE Symposium on Security and Privacy. Oakland, CA: IEEE Press, 1987.
- Covington, Michael J.; Longe, Wende; Srinivasan, Srividhya; Dey, Anind K.; Ahamad, Mustaque; Abowd, Gregory D. "Securing Context-Aware Applications Using Environment Roles." Trans. ACM. ACM SACMAT'01. Chantilly, VA, USA: ACM Press, 2002. pp. 10-20.
- Ferraiolo, David; and Kuhn, D.M. "Role-Based Access Controls." Fifteenth Annual National Computer Security Conference (NCSC). Baltimore, MD: U.S. Government, 1995.
- Fox, K, Henning, R., and Tallet, J, "IDEA – An Information Superstructure" Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX II), Anaheim, CA, 12-14 June 2001
- Hengartner, Urs, and Steenkiste, Peter. "Implementing Access Control to People Location Information." Trans. ACM. ACM SACMAT'04. Yorktown Heights, NY: ACM Press, 2004. pp. 11-20.
- Lei, Hui; Sow, Daby M.; Davis, John S.; Banavar, Gurudh; Ebling, Maria, R. "The Design and Application of a Context Service." Mobile Computing and Communications Review Vol. 6.No. 4: pp. 45-55.
- Mazalek, Ali and Davenport, Glorianna. "A Tangible Platform for Documenting Experiences and Sharing Multimedia Stories." ETP '03. Berkeley, CA USA: ACM Press, 2003. 105-09.
- Press, Microsoft. "Microsoft Encarta Dictionary for Office 2003, Windows Xp Edition". Redmond, Washington, 2004.
- Rees, Jackie, Bandyopadhyay, Subhajyoti, and Spafford, Eugene H. "Pfires: A Policy Framework for Information Security." Communications of the ACM 46.7 (2003): 101-06.
- Sandhu, Ravi. "A Logical Specification for Usage Control." Proceedings of the ninth ACM symposium on Access control models and technologies. Yorktown Heights, New York, USA: ACM Press New York, NY, USA, 2004.
- Sandhu, Ravi; Covyne, E.J; Feinstein, H.L.; Youman, C.E. "Role-Based Access Control Models." IEEE Computer Vol. 29.No. 2 (1996): pp. 38-47.

- Sandhu, Ravi; Ferraiolo, D.;Kuhn, D. "The NIST Model for Role-Based Access Control: Towards a United Standard." Trans. ACM. Fifth ACM Workshop on Role-based Access Control. Berlin, Germany: ACM Press, 2000.
- Schell, Roger R. "Computer Security -- the Achilles' Heel of the Electronic Air Force." Air University Review Vol. XXX.No. 2 (1979): pp. 16-33.
- Strembeck, Mark and Neumann, Gustaf. "An Integrated Approach to Engineer and Enforce Context Constraints in Rbac Environments." ACM Transactions on Information and System Security Vol. 7.No. 3 (2004): 392-427.
- Szilas, Nicholas. "Minimal Structures for Stories." Trans. ACM. SRMC'04. New York, NY USA: ACM Press, 2004. 25-32.
- Waraich, Atif. "Using Narrative as a Motivating Device to Teach Binary Arithmetic and Logic Gates." ITiCSE '04. Leeds, United Kingdom: ACM Press, 2004.
- Zagalo, Nelson, Barker, Anthony, Branco, Vasco. "Story Reaction Structures to Emotion Detection." Trans. ACM. SRMC'04. New York, NY USA: ACM Press, 2004. 33-38.
-