The Context and the SitBAC Models for Privacy Preservation – An Experimental Comparison of Model Comprehension and Synthesis

D. Beimel, *Member, IEEE*, and M. Peleg

Abstract—*Situation-Based Access Control* (SitBAC) is a conceptual model for representing access-control policies of healthcare organizations by characterizing situations of access to patient data. The SitBAC model enables formal representation of access situations as an ontology of concepts (Patient, Data-Requestor, EHR, Task, and Response), along with their attributes and relationships. A competing access-control model is the *Contextual Role-Base Access Control (Context)* model. The Context model uses logical expressions (rules) that specify contextual authorizations (i.e., characteristics of access requests that are available at access time). Open questions that relate to formal representation of scenarios involving access to patient data are: 1) which of the two models yields a formal representation that is easier to comprehend; 2) which of the two models facilitates the synthesis of correct models, and how does the task complexity affect performance of comprehension and synthesis. In this study, we address these questions through a controlled experiment. The results of the experiment suggest that while there are no differences between the two models when it comes to comprehending or synthesizing simple scenarios of data access, for complex scenarios there is a significant advantage to the SitBAC model, in terms of both comprehension and synthesis.

Index Terms—knowledge representation, access control, RBAC, SitBAC, authorization, conceptual model, ontology.

1 Introduction

In the information age, people are becoming more aware of the need to protect their private electronic data from falling into the wrong hands. This is especially true for medical data [1], which most people regard as sensitive. One approach to preserving privacy is through access-control mechanisms that define authorizations for data access. The most widely-used access-control mechanism is Role-Based Access Control (RBAC) [2, 3], where authorizations for data access are based on the organizational role of the data requestor. However, the role-based model is limiting, as it grants (or denies) a role-player access to data regardless of the context of the request. Yet the context of the request should often affect access control. For example, the location of the person requesting the information (e.g., at home or in the intensive care unit), the relationship between the data requestor and the patient (e.g., gynecologist or family doctor), and the time of the request could all be relevant to whether a data access request should be granted or denied. By adding contextual factors to an access request, we can more accurately express access control policies. In this way, patients' data could be released on a need-to-know basis [4], increasing the protection of patients' privacy while not compromising access to data that is essential for their care.

Motta and Furuie [5] proposed a *Contextual RBAC* model (Context model for short), which extends RBAC by constructing rules (i.e., logical expressions) that specify the contexts under which access to a patient's data should be granted or denied. The rules specify conditions that relate to context-variable values (e.g., the location of the data requestor equals the emergency room). The Context model is discussed in detail in Section 2.2.

Another access-control model that supports representation of contextual factors is our *Situation-Based Access Control* model (or SitBAC for short) [6]. SitBAC formally represents and structures situations of access to patient data as an ontology composed of concepts involved in access-control situations, their attributes and relationships. These concepts include Patient, Data-Requestor, EHR, Task, Legal Authorization, and Response. We discuss the SitBAC model in detail in Section 2.3.

Both the SitBAC model and the Context model could be used to develop formal privacy-preserving access-control policies for healthcare organizations. Both can be considered as *Conceptual Models*. However, there are differences between these two models. In terms of their structuring of contextual data-access policies, the Context model is a role-based model, where policies are structured around the role of the data requestor. On the other

- *D. Beimel is with the Department of Engineering and Management, Ruppin Academic Center, Emek Hefer, 40250, Israel. E-mail: dizzab@ruppin.ac.il.*
- *M. Peleg is with the Department of Management Information Systems, University of Haifa, Haifa 31905, Israel. E-mail: morpleg@mis.hevra.haifa.co.il.*

Manuscript received ?????????

hand, in the SitBAC model, the role is just one of the concepts comprising data-access situations. In fact, this is not even a mandatory concept, as data-request situations may ignore the role of the data requestor. For example, a situation of transferring a discharge letter from a hospital to a clinic could be defined with the data requestor being a role in the hospital or a role in the clinic, or the role may be left unspecified to generalize these two situations.

In terms of their knowledge-representation (KR) formalism, the Context model uses logical expressions, while SitBAC uses a frame-based ontology of data-request situations. According to theories of cognition, discussed in Section 2.4, an ontology, which specifies the concepts involved in access-control (AC) requests and their relations, eases cognition due to the closeness of mapping of the model to the AC domain and the ability to compose contextual AC policies from the ontological concepts in a constrained way that limits creation of errors.

Given the difference in KR between the models, we wanted to evaluate whether it would affect cognitive tasks related to use of the models. Because human beings (e.g., privacy officers in an organization) need to create and maintain access-control policies, the ease of creating and correctly comprehending the modeled policies are very important. We therefore performed a controlled experiment in which we compared the SitBAC model and the Context model in terms of the ease of defining contextual AC policies (model synthesis) and human comprehension of modeled policies (model comprehension). Furthermore, we evaluated the affect of task complexity.

The rest of the paper is organized as follows: Section 2 addresses background and related work, Section 3 presents the research hypothesis, and Section 4 presents the research design. The results of the experiment are presented in Section 5. A discussion concludes the paper.

2 Background and Related Work

We discuss the knowledge representations that we compared in this study, the cognitive aspects of comprehension and synthesis of knowledge representation, and related work on evaluation of knowledge models.

2.1 Knowledge Representation

According to Davis [7], a knowledge representation (KR) serves five roles: as 1) a surrogate to enable an entity to determine the consequences of a plan or idea; 2) a set of ontological commitments about how and what to see in the world; 3) a fragmentary theory of intelligent reasoning; 4) a medium for efficient computation; and 5) a medium for human expression.

Van Bemmel and Musen [8] discuss the different kinds of knowledge representations, including a) First-order logics; b) If…then…else rules; c) Semantic Networks; d) Ontologies (e.g., Frames, Description logics); and e) Decision-theoretic models (e.g., Bayesian Networks, Decision trees). We elaborate on ontologies, as this is the type of representation that we used in this study. An ontology [9] is an explicit, machine-interpretable specification of a conceptualization—that is, the entities, or concepts, that are presumed to exist in some area of interest, their attributes, and the relationships that hold among them. Ontology defines a common vocabulary for humans and machines that need to share information in a domain. Our reasons for developing ontologies for access-control include [10]: 1) to share a common understanding of the structure of information among people or software agents; 2) to enable reuse of domain knowledge; 3) to make domain assumptions explicit; 4) to separate domain knowledge from operational knowledge; and 5) to analyze domain knowledge.

2.2 The Context Model

The Context model [5] is an extension of the widely-used Role-Based Access Control (RBAC) model [2, 3], where logical contextual rules define conditions under which access to a patient's data is granted or denied. Fig. 1 illustrates the context model's elements and their relations in a UML class diagram. The model is defined by a tuple: <*Role, Object, Operation, Privilege-Type/Rule*>. We did not address conflict resolution in this study; hence, our examples refer to 4-tuples and do not address the authorization type element that appears in the original model.

- *Role* represents the role of the entity that seeks the access. Roles are hierarchically organized, such that authorizations can be inherited by specialized roles (e.g., a paramedic inherits the authorizations of a healthcare professional).
- *Operation* is the action that the entity wishes to execute (i.e., view or execute).
- *Object* specifies the resource to be accessed. The objects are components of electronic medical records (EHRs), organized within a part-of hierarchy structure. This structure does not express inheritance.
- *Privilege-Type* is positive when an operation is allowed, and negative otherwise. The privilege type in basic tuples may be replaced by a rule, in ruled tuples, describing contexts for which the privilege is positive.

Following is an example of a proper 4-tuple specifying the rule: "A Health-Care-Professional is allowed to view the identification section from the EHR":

```
4-tupple: <Health-Care-Professional, view, Identi-
fication-Section,+>
```

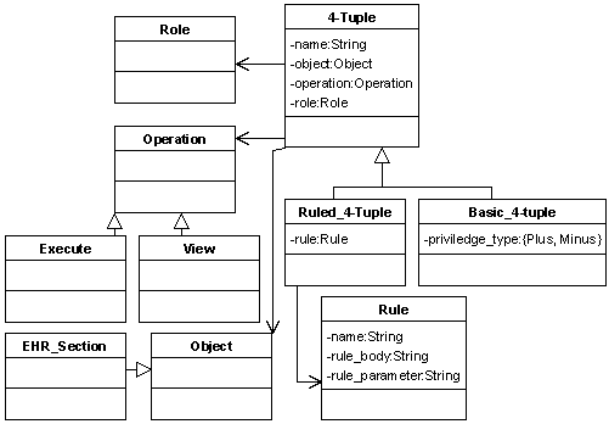

Fig. 1. The UML class diagram of the Context model

In the Context model, the privilege-type element of basic 4-tuples can be written as a *Logical Rule* that is evaluated to a Boolean, forming a Ruled 4-tuple. The rule consists of an input *Parameter* and a *Logical Expression*. A simple logical expression consists of an *Operator* (i.e., in, or, not) and one or two *Operands*. The operands can be the parameter or a Context Expression. A complex logical expression consists of two or more logical expressions connected by an operator. Context expressions consist of a context and one of its properties: `<Context-Name>.<Property-Name>`.

The expression in Fig. 2 represents a rule that consists of one parameter and one simple logical expression. The parameter is `aPatCod`, which is the code belonging to a particular patient. The context expression, written within brackets, returns true if the patient's code is included in the hospitalized-patients list (`in_patient`), otherwise it returns false. `in_patient` is a property of the Patients context (`patCtx`).

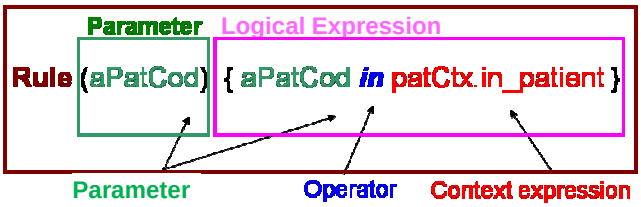

Fig. 2. A rule structure

2.3 The SitBAC Model

The SitBAC model [6] is a conceptual model that enables formal representation of detailed situations of access to patient data. Fig. 3 illustrates the model's elements and their relations via UML class diagram. The main concept within this model is the **Situation** class. Situation is characterized by a set of attributes (properties). Some attributes are simple: the *name* of the situation and its *response*, which describes the decision to accept or reject the access request situation. The other attributes belong to the following entity (component) types:

1. Data-Requestor: describes the entity asking for access to the requested data.
2. Patient: describes the subject of the requested data.
3. Task: describes the action (i.e., view, generate) on a data item from an EHR section. The action and ehr_section attributes of Task, point to Action and EHR_Section objects.
4. Relations: relations that may exist within a situation, as described below.

The complete SitBAC model includes the Legal Authorization entity, which was not addressed in this study.

The SitBAC model explicitly defines the set of properties belonging to each entity. For instance, the patient entity has the property *age*, and the data-requestor entity has the property *role*. For most properties (except for simple properties such as name and age), the possible property values are organized in hierarchical structures: Role hierarchy, EHR-section hierarchy, Action hierarchy, Location hierarchy, and Organization hierarchy.

Situation relations are divided into the following three sub-classes, which all have a *type* property.

(1) *Patient-Data_Requestor_Relation.* Such a relation indicates a permanent connection between the data requestor and the patient whose data is requested. For instance, if the data-requestor is the patient's family doctor, a relation is established between the entities whose *type* property is assigned the value `Family_Doctor_of_Patient`. A relation between the patient and the data-requestor is not mandatory and is determined based on the circumstances of the

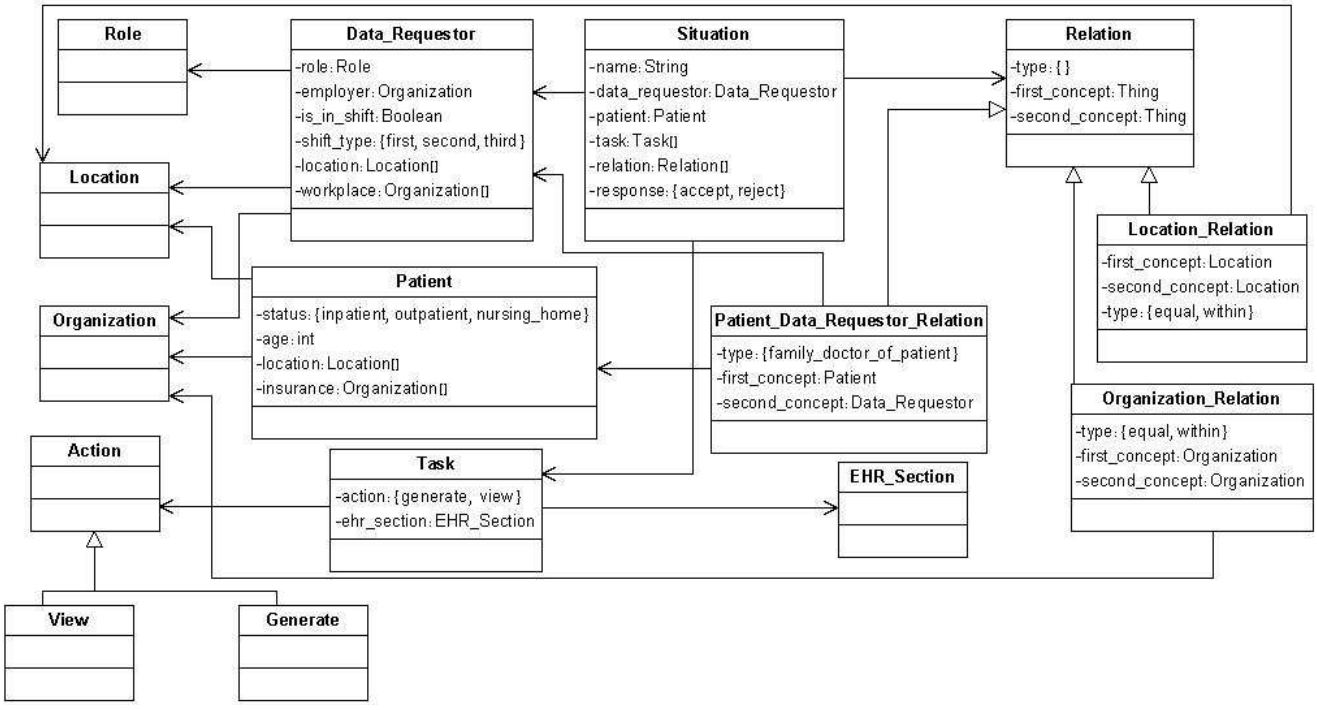

Fig. 3. The UML class diagram of the SitBAC model

situation.
(2) *Location Relation*. Such a relation represents a requirement imposed on the location properties of the patient and the data requestor. For instance, to specify that the patient and the data-requestor need to be present at the same place, we establish such a relation and set the relation type to the value `Equal`.
(3) *Organization Relation*. Such a relation represents a requirement imposed on the data requestor's employer property, the data requestor's workplace property, or the patient's insurance property. For instance, to specify that the patient must be insured by the data-requestor's employer, we establish such a relation between the Patient's insurance property and the data requestor's employer property and set the relation type to the value `Equal`.

The various scenarios of data-access requests can be expressed via **Situation Instances**, which are instances of the situation class. For each of the participating properties in a situation instance, one or both of the following two requirements must be fulfilled: 1) an appropriate value is assigned to the property (e.g., *Patient_Status* is assigned the value `inpatient` (i.e., hospitalized); and 2) the property establishes a relation with another participating property (e.g., a relationship between the patient's location and the data requestor's location).

2.4 Cognitive Aspects of Knowledge Representation

In this subsection, we present cognitive theories related to comprehension and synthesis of knowledge models, which serves as the basis for our argument that ontology is easier to use than rules. Based on this theory, we decided to develop SitBAC despite the fact that the context model had already existed.

The Context model was based on formulating contextual relationships using a textual expression language. We sought to develop an alternative conceptual contextual AC model due to two main reasons. First, we hypothesized that by examining all the concepts that are involved in AC situations, and their relationships, we could identify contexts that were not identified by the Context model. Indeed, we discovered several such contexts (e.g., Legal Authorization required for inter-organizational access) [6]. Second, we found that the textual expression language used in the context model was not easy to use for expressing contextual relationships as part of AC policies, and thought that an ontological approach would ease comprehension and synthesis of contextual AC policies. This was based on our own experience, and on the following empirical observations, and cognitive theories developed by others.

One theory concerns the closeness of the mapping from the problem world and the representation world. Green and Petre [11], discussing the representations of visual programming languages, state that "It is not easy to deal with entities in the program [representation] domain that do not have corresponding entities in the problem domain". Our ontology defines contextual policies using concepts (e.g., Patient, Data_Requestor-Patient relation, Patient's and data requestor's location) that are closer to the problem domain than the contextual RBAC model, which formulates the policies using expressions that refer to operators taken from a solution domain (e.g., net-context, security context).

A second principle concerns composition problems. Spohrer and Soloway [12] report that most of the encoding errors created by novices occur as a result of composing constructs together, rather than by lack of understanding of the constructs. In the Context model, the users do not have any guidance for assembling context variables and operators into contextual expressions. They can easily write expressions that are syntactically and semantically incorrect. On the other hand, the SitBAC ontology defines contextual policies using relations between entities. The ontology defines the inter-relations among the entities participating in relationships in an explicit and coherent way, specifying the possible relationships types, including the allowed data types of participating entities. Hence users create relation instances by assembling together legal participating entities, resulting in syntactically and semantically correct representations. The relationship-schemas help in focusing the users, such that only legal combinations of relation types and participating entities could be formed. Patel et al. [13, 14] point that comprehension is dependent on perceptual processes that focus attention and that "a function of schemata is to provide a "filtering" mechanism to experts, allowing them to selectively attend to significant information and discard irrelevant clinical information".

A third principle concerns error-proneness [11]: does the design of the notation induce 'careless mistakes'? Green and Petre [11] observe that textual programming languages (which are analogous to the expression language used in the Context model) contain a number of syntactic design features that help slips to occur or make them hard to find once they have occurred, including the need to type long identifiers and paired-delimiter system (e.g., parentheses). On the other hand, the ontological approach, supported by a convenient representation tool, such as Protégé [10], can help avoid such errors by following principles suggested by Rector et al. [15], including making the defaults the usual case, providing shortcuts for tedious tasks, and providing mechanisms to help guide users through complex tasks.

Based on the theories discussed above, we had reason to believe that SitBAC would have advantages over the Context model in terms of intellectual behavior important in problem-solving. We chose to focus on comprehension and synthesis assignments. According to Bloom's Taxonomy of educational objectives [16], synthesis requires a higher level of thinking (about the same mental model) than comprehension.

2.5 Evaluation of Knowledge Models

The goal of our study was to experimentally evaluate the usability of two conceptual models that differ in terms of their knowledge-representation formalism (logic rules against ontology concepts). We therefore searched the literature to see how other researchers have compared *conceptual modeling formalisms* in general and *knowledge*

representations in particular.

We also searched for studies evaluating the combination of logic and conceptual model. While many studies have compared formalisms based on their expressivity (e.g., ability to express concepts, axioms, relations, functions, instances—see for example [17]) or their model organization (e.g., single diagram type addressing multiple system aspects vs. multiple diagram types—see for example [18]), as judged from the specification of the formalism, we focused on empiric studies relating to model synthesis and comprehension.

TABLE 1: STUDIES ADDRESSING EMPIRICAL EVALUATION OF KNOWLEDGE MODELS

Study Reference	Formalism Compared	Formalism Type	Comparison Aspects	Study Population	Quantitative/ Qualitative	Statistical Tests Used	Repeated Measures for Validation	Cross-over Design	Measures Used to Balance Two Groups
[18]	OPM, OMT	Structural and dynamic models	Model synthesis and comprehension	88 Information Systems students	Quantitative	T-test, Wilcoxon-Mann-Whitney test, proportion difference test	+	+	1) Years of studies, 2) Previously learned course material, 3) Average homework grades
[19-21]	Entity-relationship (ER), OO, and relational	Structural models	Specification comprehension	Information Systems students	Quantitative	T-test	+	-	1) Database experience 2) Trained using one of the models
[22]	Extended ER, OO	Structural models	Correctness of specification, time to complete the task, subjects' model preference	44 students, majoring in Information Systems	Quantitative	T-test, Wilcoxon signed rank test	+	+	1) Trained to use two models, 2) Same amount of study time 3) Same lecturer
[23]	Hierarchical ER, ER	Structural models	Comprehension, time, and preference of models	42 Software Engineering students	Quantitative	T-statistic, T-test	+	+	1) Students took the same courses, 2) Students studied the two models
[24]	Sequence diagrams, collaboration diagrams, state diagrams	Dynamic model	Semantic comprehension	18 last-year students of Informatics		ANOVA (using the F- distribution)	+	+	NA
[25]	Hierarchical, network, and relational DB models	Data models and data language	Correctness, required time for coding, and required time for debugging	41 Computer Science students	Quantitative	Average and percent calculations	+	-	1) Previous knowledge of APL, 2) Previous work experience as programmers
[33]*	Novice using EZ-Protégé Axiom Language (Easy-PAL) vs. Expert using PAL	First Order Logic (PAL) with/out ontology templates	Correctness of synthesis, time for synthesis	7 novice Protégé users and one expert	Quantitative	Average and Percent calculations	+	NA	-
[32]*	UML vs. UML+ Object Constraint Language (OCL)	Structural and dynamic models with/out constraint language	Error detection, model comprehension, facilitation of change management	38 Computer-Science students trained in UML	Quantitative	paired T-test, F-test, Wilcoxon test	+	+	1) Grades in previous UML course, 2) Previously learned OCL 3) Engineering program

***reviewed in the Discussion section**

The related studies are summarized in Table 1. Many studies addressed structural [19-23] and behavioral [18, 24] system models and database conceptual models [25][25]. Usually, the aspects compared were the correctness of synthesized models, the time required for synthesis, and the accuracy with which models were comprehended.

The studies often used students as subjects and involved quantitative evaluation. Sjøberg et al. [26] conducted a survey of controlled experiments in software engineering drawn from the 5,453 scientific articles published in 12 leading software engineering journals and conferences during 1993 through 2002. They found that103 articles (1.9%) reported controlled experiments in which individuals or teams performed one or more software engineering tasks. Only 15 out of 103 involved design, including notation, documentation, methodologies and representation. This low number may explain the fact that we could not find a study similar to our study, where two knowledge representations were compared in terms of model comprehension and synthesis.

3 Research Hypothesis

As noted, the main differences between the SitBAC model and the Context model lie in how they structure and represent the context of data requests. Our null hypothesis (H_0) was that there is no difference between the two models in any aspects of model-comprehension and model-synthesis.

To this end, we designed and performed an experiment comparing participants' ability to comprehend and express scenarios of access to patient data via the Context and SitBAC models.

4 Research Design

Our objective was to assess whether, following the theories of closeness of mapping, composition, and error-proneness (see Section 2.4), the SitBAC model, which expresses all the entities and relationships involved in an AC request explicitly, would ease *synthesis* of AC scenarios, or even the less difficult process [16] (see Section 2.4) of their *comprehension*, as compared to the Context model. Moreover, we wanted to assess the affect of task complexity on comprehension and synthesis.

In order to eliminate any confounding effects from the environment (i.e., the tools in which the models are implemented), we implemented both models using the same environment. We carried out the experiment as a mid-term exam in a "Knowledge Representation and Decision-Support" course, described below. Subsection 4.1 presents the research population, while subsection 4.2 describes the experimental settings. Subsection 4.3 discusses the models' implementation. Subsection 4.4 discusses the modeling comprehension and synthesis assignments. Subsection 4.5 discusses task complexity. The last two subsections (4.6-4.7) present the two parts of the experiment: model comprehension and model synthesis.

4.1 Population Background and Training

The students who participated in the experiment were students enrolled in Knowledge Representation and Decision-Support–an advanced elective course offered to third-year students of the Management Information Systems (IS) program at the University of Haifa.

One could think that the suitable population for such an experiment would be a group of clinicians. However, we [27] as well as other research groups [28] who have worked on representing clinical knowledge in a computer-interpretable format have found that clinicians are not the right users to represent the knowledge formally. Rather, knowledge engineers, also known as information system analysts, are the appropriate target groups. These researches have recommended that clinicians would work together with information system analysts in validating the encoded knowledge. In the AC domain, the respective users are privacy officers. The population used in our experiment (IS students) closely match privacy officers in terms of their education and training; however, students are not as experienced in representing AC as privacy officers.

The students who participated in the experiment have previously taken the following related courses: Information Systems Design; Information Systems Analysis; and Discrete Mathematics and Algorithms. The Knowledge Representation and Decision-Support course covers, among other topics, such knowledge-representation methods as rule bases, semantic networks, taxonomies, frames, ontologies, and logic. Some of the classes were conducted in a computer lab, where the students learned how to use software tools such as Ruby [29] and Protégé [30]. Most of the examples for knowledge-based decision-support systems were taken from the domain of medical informatics. The instructor of the course (the second author) taught the entire course, except for the classes where the students presented their projects.

The Context model and the SitBAC model were presented to the class after the students studied frame-based ontologies, had a tutorial on the Protégé tool in the computer lab, and completed a homework assignment in which they developed an ontology using Protégé. The Context and SitBAC models were presented to the class by the first author through 30-minute PowerPoint presentations, which included the models' definitions, their Protégé implementations, and examples of modeling scenarios of access to patient data.

The experiment took place during 2007. The course population consisted of the entire class of students, to address the threat of aptitude-treatment-interaction. The 16 students included 15 undergraduates in their third to fifth year of study and 1 graduate student.

4.2 Experimental Settings

A week after the presentations, the experiment was carried out as a mid-term exam. The class was divided into two groups, where group A received a SitBAC model exam while group B received a Context model exam. Each group consisted of 8 students. The students were

paired in advance, to reduce confounding factors and were randomly assigned to one of the two groups. The students were matched with respect to the following relevant extraneous variables that tracked their background and skills, including a) their year of study (3, 4, or 5/graduate study); b) whether they had a second major and what it was (computer science or economics); c) their current grade-point average (75-92); d) their performance level on homework assignments (high, medium, or low); and e) their class attendance.

The exam was carried out in a quiet computer lab to minimize random irrelevancies in the setting and prevent diffusion between participants. Two Protégé projects (one for SitBAC and one for the Context model) were created in advance by the experimenters and were installed on the computers in the lab. Each student received an identical textual exam and access to one of the projects, depending on his group assignment.

The textual exam included four parts: 1) a description of the model (SitBAC or Context); 2) six textual descriptions of data-access scenarios, for each of which we created ontology instances in the attached Protégé project; 3) a short tutorial that instructed the students to study the six textual descriptions and their ontology instances and explained how to re-create an ontology instance for one of them; and 4) three assignments that the students had to complete. Both sets of exams included the same six textual descriptions and the same assignments. Some of the context rules that appeared in the exam were taken from Motta and Furuie [5].

The Protégé projects included the ontology of the corresponding models, along with nine instances. Six instances were created for the tutorial, and three were created for the specification comprehension assignment.

We allocated an hour and a half for the test. We measured the time that each student needed to complete each part of the exam.

4.3 Implementation of the Models

As the SitBAC model was implemented via a frame-based ontology, to limit threat to internal validity, we decided to represent the Context model via a frame-based ontology as well, keeping the representation of the contextual rules as textual expressions. We chose the Protégé [30] knowledge-modeling tool and used it to create a frame-based ontology for each model.

The ontology for the Context model includes five classes, which correspond to the 4-tuple concepts described in Fig. 1 (Role, Operation, Object, and Privilege_Type/Rule), where the fifth class, Rule, is used to express contextual privilege types. In addition, we created an abstract 4-tuple class, with two concrete subclasses: Basic_4_tuple that use the basic privilege type and Ruled_4_tuple, where the privilege type is extended by a rule. As in the SitBAC model, hierarchies for allowed values of the Role, Operation, and Object (EHR_Section) concepts were created. The scenarios of data-access requests are expressed as instances of the two 4-tuple classes. Fig. 4 represents the following scenario: "Nutritionist is allowed to view diagnosis if the patient is inpatient, the location of access is within the emergency domain, and the time of access is 19:00-7:00".

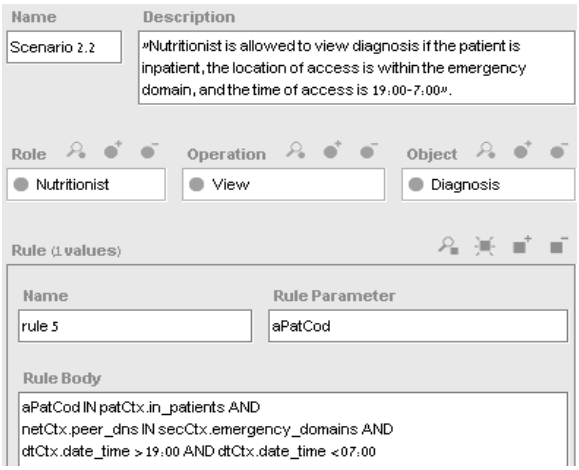

Fig. 4. An instance of the Context-model in Protégé

Values for the *role*, *operation*, and *object* slots are selected by pointing to existing concepts from the appropriate hierarchies in the ontology: the Role, Operation, and Object hierarchies. Thus, `Nutritionist` is a subclass in the Role hierarchy, `View` is a subclass of the Operation hierarchy, and `Diagnosis` is a subclass in the Object hierarchy. The rule property points to a (nested) instance of the Rule class. This instance includes the properties *name*, *rule parameter*, and *rule body*. Note that contextual-rule expressions specified in the rule body have to be expressed as text strings, thus keeping the representation of the contextual rules as in the original Context model.

The *rule_body* specifies the rule:

```
aPatCod IN patCtx.in_patients AND netCtx.peer_dns
IN secCtx.emergency_domains AND dtCtx.date_time >
9:00 AND dtCtx.date_time < 07:00
```

Its corresponding textual description is: "the patient is an inpatient, the location of access is the emergency domain, and the time of access is 19:00-7:00."

The SitBAC ontology corresponds to the model's elements described in Fig. 3. Fig. 5 shows a situation instance describing the following policy: "Audit physician is allowed to view diagnosis of her inpatient who is not insured by the audit physician's employer, if the audit physician is in the emergency room".

As shown in Fig. 5, the *data_requestor*, *patient*, *task*, and *relation* properties of the situation instance point to instances of the data-requestor, patient, task, and relation classes, and the response property is set with value `accept`. These nested instances also contain properties. For example, the patient's *status* property is assigned the value `Inpatient`. Some properties point to classes in the ontology's class hierarchies (Role hierarchy, EHR-section hierarchy, Action hierarchy, Location hierarchy, and Organization hierarchy). For example, the *role* property of the data requestor points to the `Audit_Physician` Role subclass and the data requestor's *location* property points to the `Emergency_Room` Location subclass. The situation's *task* has two properties: the *action* property, which points to the Action subclass `View`, and the *EHR section* proper-

ty, which points to the EHR_Section subclass `Diagnosis`. We created one property-to-property relation instance: `Patient_Insurance not-equal Data_Requestor_Employer`.

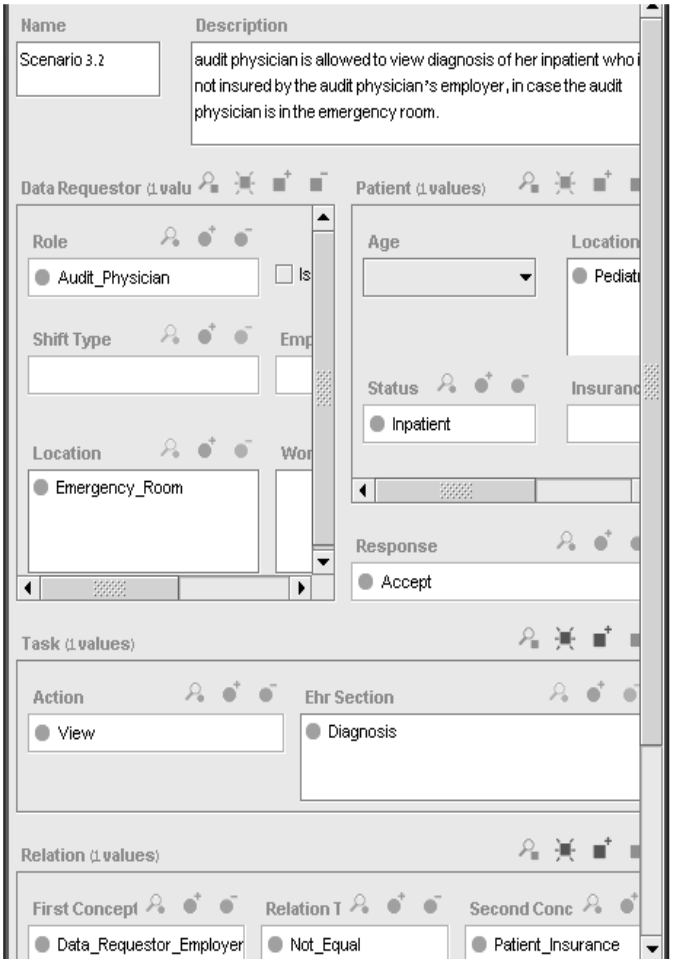

Fig. 5. A SitBAC situation instance created via Protégé

4.4 Comprehension and synthesis assignments

The first of the three assignments in the exam addressed model comprehension, while the second and third assignments addressed model synthesis, as follows:

1. *Comprehension*. The first assignment was to view three access-request scenarios that were represented as ontology instances (of the SitBAC or the Context model) of the respective Protégé project. The students were asked to describe the scenarios textually.
2. *Synthesizing model instances using existing values*. In the second assignment, students were presented with textual descriptions of three new scenarios. The students were asked to create for each scenario a corresponding instance in the corresponding Protégé project. Though new instances of situation or 4-tuple had to be created, all the required values already existed in the Protégé project from the six tutorial examples and the three examples of assignment 1. Thus, to accomplish this assignment, the students had to provide slot values for situation or 4-tuple by selecting existing values.
3. *Synthesizing model instances while creating new values*. This assignment was similar to the second: it included a description of three scenarios and, again, the students were required to create for each of these scenarios an instance of situation or 4-tuple. However, several values needed for completing the assignment were not defined in the ontology, and the students had to create them (e.g., the relationship shown on the bottom of Fig. 5, Data_Requestor's employer not-equal Patient's insurer was new).

Synthesizing instances while creating new values is a more complicated assignment than synthesizing instances using existing values. We wanted to distinguish between these two assignments as it was possible that there would be a difference between the SitBAC and Context models in assignments that required creating new values but not in assignments that used existing values.

4.5 Task Complexity

In our experiment, the students had to complete three assignments. We decided to break each assignment into several tasks, in order to assess the difference in performance between the two groups based on more than one exercise (prevent mono-operation bias). The process of breaking each assignment into tasks is described in detail in subsections 4.6 and 4.7. We categorized tasks into *Simple* and *Complex* comprehension/synthesis tasks. Following the relationship between the coherence of text and its degree of connectedness [13, 14], simple tasks manage single propositions that involve a single concept (e.g., the role of the data requestor is a nutritionist), whereas complex tasks involve modeling an idea that refers to more than one concept and involves at least two propositions (e.g., the family doctor is requesting the data of a patient of hers). Hence simple tasks involve properties that in both the SitBAC and Context models receive values by pointing (see section 4.3) to existing classes (e.g., the *status* property of Patient points to the Status subclass `Inpatient`, in Fig.5) or instances (e.g., the *task* property points to the Task instance `View.Diagnosis`, in Fig.5).

Since the simple tasks were identical in the Context and the SitBAC models, they served as an internal control for the experiment; the performance of the two groups with respect to such simple tasks should not differ. Complex tasks involve relationships between contextual properties of a situation or a 4-tuple. In the Context model, they are expressed using textual rules (e.g., the content of the *rule_body* in Fig.4) In the SitBAC model, they are represented as instances of the Relation subclasses (e.g., the *relation* property points to the instance `Patient_Insurance:not-equal:Data_Requestor_Employer` in Fig.5), which, as explained in Section 2.4, provides a schemata that focuses the modeler and helps him in creating syntactically and semantically correct relations. In addition, pointing to existing instances that appear in different tasks provides a short-cut for a tedious task [15], facilitating reuse of specifications that have already been validated.

4.6 Model Comprehension (Assignment 1)

The purpose of this part of the experiment was to compare the SitBAC model with the Context model with respect to the students' ability to understand instances created in each model. In this subsection we describe how we assessed comprehension from the textual descriptions provided by the students for the Context and the SitBAC model instances.

Fig. 4 shows an instance of the Context model. The instance expresses one of the scenarios in assignment 1, and is explained in Section 4.3.

In order to analyze comprehension, we partitioned assignment 1 into several tasks. In each task, we expected the students to textually indicate one fact (e.g., indicate that the data requestor's role is nutritionist). When a student successfully accomplished a task, he received one point, otherwise he received zero points. Following is a list of tasks for assignment 1. Some of the tasks appear in all three scenarios of assignment 1, and some appear only in some scenarios (as specified in Appendix A).

1. Indicate the data-requestor's *role* (e.g., `Nutritionist`, shown in Fig. 4)
2. Indicate the *response/privilege type* (e.g., `accept`, shown in Fig. 5)
3. Indicate the *action*/operation (e.g., `View`, in figures 4 and 5)
4. Indicate the *EHR-section* (e.g., `Diagnosis, in figures 4 and 5`)
5. Indicate the data-requestor's *location-of-access* (e.g., `peer_dns IN secCtx.Emergency_domains` – see rule_body in Fig. 4)
6. Indicate the data-requestor's *time-of-access*, as in the rule_body of Fig. 4
7. Indicate the patient's *status* (e.g., `aPatCod IN patCtx.in_patients`, as in the rule_body of Fig. 4)
8. Indicate the relation between the patient and the data-requestor (e.g., `Patient_Insurance not-equal Data_Requestor_Employer`, as in the bottom of Fig. 5.

Attending to construct validity, we used our definitions of simple and complex tasks to categorize tasks 1-4 as simple and tasks 5-8 as complex, and analyzed them as two groups with repeated measures. The three scenarios included 12 simple tasks and 6 complex ones (see App. A). Hence, the maximum score that each student could get was 12 for simple comprehension and 6 for complex comprehension.

4.7 Model Synthesis (Assignments 2 and 3)

In assignments 2 and 3, we compared the SitBAC and Context models with respect to correctness of model synthesis. Fig. 5 shows an instance of the SitBAC model that expresses a scenario of assignment 2, which is explained in Section 4.3.

Similarly to the analysis of model comprehension, assignments 2 and 3 were partitioned into several tasks. In each task, we expected the student to express, via model elements, one textual fact (e.g., to assign the `nutritionist` Role subclass to the data-requestor's *role* property, as shown in Fig. 4). When a student successfully accomplished a task, he received one point, otherwise he received zero points. Note that the students were scored against their ability to fulfill a task. Thus, in the Context model, it could be that while the entire expression was not well-defined, the student obtained a partial score for tasks that he accomplished.

We identified 14 tasks. Each of these tasks was examined from two aspects: (a) simple or complex (as in assignment 1); and (b) existing or new. As explained in Section 4.4, 'new' stands for a new value that had to be created as a class in the ontology in order to accomplish the task, while 'existing' stands for a value that already existed in the Protégé project. Based on these aspects, each task was categorized into one of four categories:

1. *Simple-Existing.* This category includes four tasks: (1) express data-requestor's role using an existing class from the Role hierarchy; (2) express response by selecting an existing value of `accept` or `reject`; (3) express an action/operation, using an existing class (e.g., `View`) from the Action hierarchy; and (4) express an EHR section/object using an existing class from the EHR_Section hierarchy. The maximum score for this category was 15 points as these 4 tasks appeared several times in the six scenarios of tasks 2 and 3 (see Appendix A).
2. *Complex-Existing.* This category includes three tasks: (1) express data-requestor's location of access using an existing instance (or existing context property – see section 2.2). For example, point to the existing instance of a rule-body that includes the expression `netCtx.peer_dns IN secCtx.emergency_domains`; (2) express patient's status (e.g., `inpatient`) using an existing instance or context property. For example, point to the existing instance of a rule-body that includes the expression `aPatCod IN patCtx.in_patients`; and (3) express relation between patient and data-requestor using existing instances or context properties (e.g., by pointing to a relation instance, such as that shown in the bottom of Fig. 5). The maximum score for this category was 6 points.
3. *Simple-New.* This category includes two tasks: (1) express EHR section/object by creating a new EHR section subclass; and (2) express data-requestor's role by creating a new subclass in the Role hierarchy. The maximum score for this category was 6 points.
4. *Complex-New.* This category includes five tasks: (1) express data-requestor's location of access by creating a new instance or defining a new context property. For example, define the location of primary care clinic and use it to express the location of the family doctor; (2) express data-requestor's work-shift by creating a new instance (e.g., express the fact that the data requestor can have access to data when he is working his shift); (3) express data-requestor's workplace by creating a new instance; (4) express patient's location by creating a new instance. For example, define the workplace of `Pediatrics Dept.` and use it to express the workplace of the physician; and (5) express relation between the patient and the

data-requestor by creating new relation instances (e.g., the patient needs to be insured by the family doctor's employer). The maximum score for this category was 6 points.

Assignment 2 included one bonus question (see 3.3 in Table 14 in the Appendix). 4 of 8 students from the Context-model group preferred to skip this question, while only one student from the SitBAC-model test skipped this question. Accordingly, we did not award points for tasks contained in the bonus question.

As another control, we examined the students' ability to use Protégé, which we judged by two tasks: (1) add a new concept to the ontology (e.g., `Registered Nurse` added to the Role hierarchy; and (2) position the new concept in the correct place in the concept hierarchies (e.g., `Registered Nurse` as a subclass of `Nurse`). The maximum score for this category was 8 points.

The detailed scenarios, their tasks, and maximum scores are provided in Appendix A. We used the Wilcoxon matched-pairs signed rank test [31] to compare the achievements of the two groups.

5 Results

In this section, we present the statistical results. Note that the Wilcoxon test works only when N ≥4.

5.1 Model Comprehension

Table 2 summarizes the results of the simple comprehension category. The maximum score is 12 points (for four tasks). Since N=1, our data is consistent with the null hypothesis H_0, which states that the groups' simple comprehension is not dependent on the chosen model (SitBAC or Context).

TABLE 2:
RESULTS FOR SIMPLE COMPREHENSION

pair #	SitBAC	Context	di	rank-of-d
A	12	12	0	--
B	12	12	0	--
C	12	12	0	--
D	12	12	0	--
E	12	12	0	--
F	12	12	0	--
G	12	11	1	1
H	12	12	0	--
N= 1				

For each pair (from the 8 pairs A...H), one person used the Context model and the other the SitBAC model. Their scores are reported in the table, along with di - the difference score for the pair, and rank-of-d - the rank of the difference score (di) and the sign of the difference. Below the table, we report N – the number of pairs with d ≠0.

Table 3 presents the results of the complex comprehension category. With α=0.0156, our data is not consistent with the null hypothesis that there is no difference between the models in that category.

5.2 Model Synthesis

We present the results for model synthesis in the four categories discussed in section 4.5.

TABLE 3:
RESULTS FOR COMPLEX COMPREHENSION

pair #	SitBAC	Context	di	rank-of-d
A	6	5	1	3
B	6	5	1	3
C	6	5	1	3
D	6	4	2	6.5
E	6	5	1	3
F	5	4	1	3
G	6	4	2	6.5
H	6	6	0	ignored
N= 7, T+=28, T-=0 →α=0.0156				

Below the table, we report N – the number of pairs with d ≠0, T+ - the sum of the positive di's, T- - the sum of the negative di's.

Table 4 shows that our data is consistent with the null hypothesis that there is no difference between the two models in regard to synthesis of simple existing instances.

TABLE 4:
RESULTS FOR SYNTHESIS OF SIMPLE EXISTING

pair #	SitBAC	Context	di	rank-of-d
A	15	15	0	ignored
B	15	15	0	ignored
C	15	13	2	3.5
D	15	15	0	ignored
E	15	13	2	3.5
F	15	15	0	ignored
G	15	14	1	1.5
H	15	14	1	1.5
N= 4, T+=10, T-=0 →α=0.125				

Table 5 displays the results for synthesis of instances belonging to the complex existing category. As α=0.0313, our data is not consistent with the null hypothesis that there is no difference between the models in creating complex expressions based on existing values.

TABLE 5:
RESULTS SYNTHESIS OF COMPLEX EXISTING

pair #	SitBAC	Context	di	rank-of-d
A	6	6	0	Ignored
B	6	6	0	ignored
C	6	5	1	1.5
D	6	4	2	4
E	6	4	2	4
F	6	4	2	4
G	6	0	6	6
H	6	5	1	1.5
N= 6, T+=21, T-=0 →α=0.0313				

Table 6 shows our data is consistent with the null hypothesis that there is no difference between the models in creating simple expressions based on new values.

Table 7 summarizes the results in the last category: complex-new. As α=0.0078, our data is not consistent with the null hypothesis that there is no difference between the models in expressing complex instances based on values that have to be created first.

TABLE 6:
RESULTS FOR SYNTHESIS OF SIMPLE NEW

pair #	SitBAC	Context	di	rank-of-d
A	5	5	0	--
B	5	5	0	--
C	5	5	0	--
D	5	5	0	--
E	5	4	1	1
F	5	5	0	--
G	4	4	0	--
H	5	5	0	--
N= 1				

TABLE 7:
RESULTS FOR SYNTHESIS OF COMPLEX NEW

pair #	SitBAC	Context	Di	rank-of-d
A	6	1	5	6.5
B	4	3	1	1
C	6	2	4	3.5
D	5	0	5	6.5
E	6	1	5	6.5
F	6	2	4	3.5
G	5	0	5	6.5
H	6	3	3	2
N= 8, T_+=36, T_-=0 $\rightarrow\alpha$=0.0078				

TABLE 8:
SUMMARY OF THE RESULTS FOR CONCEPT ADDITION

pair #	SitBAC	Context	di	rank-of-d
A	5	5	0	ignored
B	5	6	-1	-2
C	5	5	0	ignored
D	6	5	1	2
E	6	4	2	4.5
F	5	6	-1	-2
G	5	3	2	4.5
H	6	6	0	ignored
N= 5, T+=11, T-=4 $\rightarrow\alpha$=0.4375				

TABLE 9:
SUMMARY OF THE RESULTS FOR THE THREE ASSIGNMENTS

	Assignment 1 Comprehension		Assignment 2 Creating Existing		Assignment 3 Creating New	
	Simple	Complex	Simple	Complex	Simple	Complex
α Values	---	α=0.0156	α=0.125	α=0.0313	---	α=0.0078
Difference in Favor of:	---	SitBAC model	---	SitBAC Model	---	SitBAC model

The students' ability to use Protégé (create new concepts and insert them correctly into the class hierarchy) is shown Table 8. Our data is consistent with the null hypothesis that there are no differences between the groups. Table 9 summarizes the results of the entire experiment.

Though we allocated an hour and a half for the test, we allowed additional time for students who did not complete their assignments. Several students utilized this extra time, requiring up to 30 additional minutes. We measured the time each student needed to complete his assignments, and used a paired T-test to compare the results of the Context–model group and the SitBAC-model group. Table 10 summarizes the results. As the table shows, the time needed to complete assignment 1 (model comprehension) was significantly shorter for the SitBAC model. The results for assignment 2 and 3 (model synthesis) do not show a significant difference between the two models. The results for assignment 2 are inconclusive since the third part of assignment 2 was defined as a bonus question; four of the eight students who used the Context model, and one of the eight students using SitBAC, preferred not to answer this question.

TABLE 10:
SUMMARY OF THE RESULTS OF THE TIME DIMENSION

	Assignment 1 Comprehension		Assignment 2 Synthesis Existing		Assignment 3 Synthesis New	
Model	Context	SitBAC	Context	SitBAC	Context	SitBAC
Average time (minutes)	25.87	11.87	26.75	28.75	43.12	45.25
t-test	4.32		-0.320		-0.453	
P-values	0.0035		0.7581		0.6641	
Difference in favor of:	SitBAC model		--- (inconclusive)		---	

6 DISCUSSION

We reported the results of a controlled experiment designed to compare the usability of the Context and SitBAC models. The models were compared in terms of two constructs: 1) modeling assignment: model comprehension and model synthesis (using existing or new values); and 2) task complexity. Taking into account that simple tasks (in all three types of modeling assignments) were technically done exactly the same in the Context and SitBAC model, we were not expecting to see a difference in performance between the models in simple tasks. As shown in Table 9, the results of all the simple tasks in the experiment showed no difference between the two models. The simple tasks included slots that involved no contextual variables and that were represented in both ontologies in an identical way. Thus, they served as a control showing that indeed, the two groups of students were equally able to perform the assignments. The second control showed that both groups were equally good at using the Protégé tool.

Based on the theories of closeness of mapping, composition, and error-proneness (see Section 2.4), we thought that due to the ontological design of SitBAC, complex tasks would be easier to perform using the SitBAC model, as compared to the Context model. We expected to see a difference in performance at least in the hardest tasks of model synthesis using new values, and perhaps also in simpler tasks of synthesis using existing values and comprehension (see Section 4). The results of the complex cases showed that comprehension and synthesis of access-control scenarios (whether using existing

values or creating new ones) was significantly better when using the SitBAC model.

Considering construct validity, the experiment's results provided evidence that supports our theoretical view of the relations among our constructs. Measures that theoretically are supposed to be highly interrelated are also highly interrelated in practice (convergent validity); in our experiment, all simple tasks (comprehension and synthesis using existing or new values), which are technically done in an identical way in the Context and SitBAC models show no difference between the two models. Similarly, measures that should not be related to each other are not related in practice (discriminate validity); in our experiment, complex tasks (which require reasoning with relations, which are represented in the SitBAC model in a way that focuses attention and permits only syntactically and semantically correct relations to be specified) were done better using the SitBAC model than the Context model. The results are consistent with our theory that it is difficult to understand and to express a scenario of data-access via complex logical expressions, as in the Context model. It is easier to understand and express scenarios using the SitBAC model, where the same information is expressed by selecting concepts from constrained concept hierarchies.

6.1 Critical Review of the Experiment Setting

The experiment presented in this paper has several threats to validity.

(1) Threats to external validity concern the generalization of the results to the population of intended uses and the settings. The related threats include: (a) though the target population (IS students) closely match privacy officers, in terms of their training, the chosen population may not generalize to experienced privacy officers. In addition, the environment was a classroom exam rather than the privacy officer's environment; (b) the number of students participating in the experiment was only 16. Although the statistical test took account of this, some samples contained too many ties, so that the difference between scores was small and its significance could not be determined; and (c) the study population included students who were not experienced with the two formalisms used in the experiment. Further training may enable users to overcome the learning curve required for using logical expressions. Indeed, this is suggested by a study [32] that experimentally evaluated whether using the Object Constraint Language (OCL) (a logical language) in addition to UML (an intuitive graphical formalism) has an impact on software engineering activities carried out by fourth-year computer and software engineering students.

(2) Threats to internal validity concern the causal inferences in scientific studies. The related threats include: (a) although the students studied the two models in the tutorial, which they read individually and unassisted during the test, they had seen 10-minute presentations of each of the two models during class, a week prior to the test. These presentations were not given by the same person; the first author, who is one of the developers of the SitBAC model, presented the SitBAC model to the class; the Context model was presented to the class by a student; and (b) the analysis of the time dimension for model synthesis using existing values was distorted by the bonus question in assignment 2. Therefore, this does not rule out possible differences between the models in this dimension.

(3) Threats to construct validity concern the assessment of translating the ideas or theories into actual measures. The two models were implemented as a frame-based ontology. While this is the native implementation of the SitBAC model, the Context model is usually expressed entirely in text. Since the logical rules still needed to be expressed in text in the current experimental setting, we do not expect to obtain different results for a text-only implementation of the Context model. However, we cannot rule this out.

6.2 Future Experiments

The current experiment was carried out by students who were in their third year of studies. An interesting future version of the experiment could include participants who would normally be the ones implementing AC policies in their organizations, such as professional database administrators or knowledge engineers.

The SitBAC model presented in this study contains a *Legal Authorization* entity that is used to represent scenarios where the data requestor seeks information owned by organizations other than his own. In such cases, a legal form authorizing data access needs to be present. Examples of SitBAC instances using the Legal Authorization may be found at http://www.technion.ac.il/~dizza/IndirectAccessRequestsSituations.doc. Inter-organization scenarios are more complex than intra-organization scenarios. It would be interesting to compare the SitBAC model against the Context model in terms of comprehension and synthesis of inter-organization scenarios.

Other researchers [33] have addressed the difficulty of *integrating logical expressions into an ontology*. This occurs when users cannot translate their thoughts into abstract and symbolic representations. To this end, the authors developed an axiom-modeling methodology, called EZ-PAL, whereby they identified templates based on structural repetitions among existing axioms. In EZPAL, a plug-in for the Protégé ontology editor, the templates are available as fill-in-the-blank English sentences for creating logical expressions in the Protégé Axiom Language (PAL) as part of their ontology instances without previous knowledge of the underlying axiom. The authors have shown that novice Protégé users performed with the EZPAL tool almost as well as an expert who had extensive experience composing PAL axioms without using templates. Further analysis of the results shows that the expert spent a significant amount of time checking the syntax and logic of the axioms. Developing such templates for the logical rules of the Context model can potentially improve its usability.

REFERENCES

[1] R. Morgan, "Community Attitudes to Privacy," Office of the Australian Federal Privacy Commissioner 2001.

[2] R. S. Sandhu, E. J. Coyne, and C. E. Youman, "Role-Based Access Control Models," *IEEE Computer*, vol. 29, no. 2, pp. 38-47, 1996.

[3] R. Sandhu, "The NIST Model for Role-Based Access Control: Toward a Unified Standard," Proc Fifth ACM workshop on role-based access control, 2000.

[4] J. B. D. Joshi, W. G. Aref, A. Ghafoor, and E. H. Spafford, "Security Models for Web-based Applications," *Commun ACM*, vol. 44, no. 2, pp. 38–44, 2001.

[5] G. Motta and S. Furuie, "A Contextual Role-Based Access Control Authorization Model for Electronic Patient Record," I*EEE Trans Info Tech in Biomedicine*, vol. 7, no. 3, pp. 202-7, 2003.

[6] M. Peleg, D. Beimel, D. Dori, and Y. Denekamp, "Situation-Based Access Control: Privacy Management via Modeling of Patient Data Access Scenarios," *J Biomed Inform*, vol. 41, no. 6, pp. 1028-40, 2008.

[7] R. Davis, H. Shrobe, and P. Szolovits, "What is a Knowledge Representation?", *AI Magazine*, vol. 14, no. 1, pp. 17-33, 1993.

[8] J. H. van Bemmel and M. A. Musen, Handbook of Medical Informatics: Springer, 1997.

[9] T. R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *Intl J Human-Computer Studies*, vol. 43, no. 5-6, 1995.

[10] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford Medical Informatics Technical Report SMI-2001-0880 2001.

[11] T. R. G. Green and M. Petre, "Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework," *J Visual Lang & Comput*, vol. 7, no. 2, pp. 131-74, 1996.

[12] J. C. Spohrer and E. Soloway, "Novice mistakes: are the folk wisdoms correct? Studying the Novice Programmer," *Commun ACM*, vol. 29, no. 7, pp. 624-32, 1986.

[13] V. L. Patel, J. F. Arocha, M. Diermeier, E. H. Shortliffe, and R. A. Greenes, "Methods of Cognitive Analysis to Support the Design and Evaluation of Biomedical Systems: The Case of Clinical Practice Guidelines," *J Biomed Inform*, vol. 34, no. 1, pp. 52-66, 2000.

[14] V. L. Patel, J. F. Arocha, and D. R. Kaufman, "A primer on aspects of cognition for medical informatics," *J Am Med Inform Assoc*, vol. 8, no. 4, pp. 324-43, 2001.

[15] A. L. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe, "Designing User interfaces to Minimise Common Errors in Ontology Development: The CO-ODE and HyOntUse Projects," Proc. UK E-Science All Hands Meeting, Nottingha, 2004.

[16] B. S. Bloom, T*axonomy of Educational Objectives, Handbook 1: Cognitive Domain*. Addison Wesley, 1956.

[17] A. Gómez-Pérez and O.Corcho, "Ontology Languages for the Semantic Web," *IEEE Intel Sys*, vol. 17, no. 1, pp. 54-60, 2002.

[18] M. Peleg and D. Dori, "The Model Multiplicity Problem: Experimenting with Real-Time specification Methods," *IEEE T software eng*, vol. 26, no. 8, pp. 742-759, 2000.

[19] P. Shoval and I. Frumermann, "OO and EER Conceptual Schemas: A Comparison of User Comprehension," *Database Management,* vol. 5, no. 4, pp. 28-38, 1994.

[20] D. Batra, J.A. Hoffer, and R.P. Bostrom, "Comparing Representations with Relational and EER Models," *Comm. ACM*, vol. 33, no. 2, pp. 126-139, 1990.

[21] P. Palvia, C. Liao, and P.L To, "The Impact of Conceptual Models on End-User Performance," J *Database Management*, vol. 3, no. 4, pp. 4-15, 1992.

[22] P. Shoval and S. Shiran, "Entity-Relationship and Object-Oriented Data Modeling-an Experimental Comparison of Design Quality," *Data and Knowledge Eng*, vol. 21, no. 3, pp. 297-315, 1997.

[23] P. Shoval, R. Danoch, and M. Balaban, "Hierarchical Entity Relationship Diagrams - The Model, Method of Creation and Experimental Evaluation," *Requirements Eng J*, vol. 9, no. 4, pp. 217-228, 2004.

[24] M.C. Otero and J.J. Dolado, "An Initial Experimental Assessment of the Dynamic Modelling in UML," *Empirical Software Eng J*, vol. 7, no. 1, pp. 27–47, 2002.

[25] F.H. Lochovsky and D.C. Tsichritzis, "User performance considerations in DBMS selection," *Proc ACM SIGMOD,* pp. 128-134, 1977.

[26] I.K. Sjøberg, J.E. Hannay, O. Hansen, V. Kampenes, et al. "A Survey of Controlled Experiments in Software Engineering," *IEEE T Software Eng*, vol. 31, no. 9, pp. 733-753, 2005.

[27] M. Peleg, D. Wang, A. Fodor, S. Keren, and E. Karnieli, "Lessons learned from adapting a generic narrative diabetic-foot guideline to an institutional decision-support system," S*tud Health Technol Inform*, vol. 139, pp. 243-52, 2008.

[28] E. Shalom, Y. Shahar, E. Lunenfeld, M. Taieb-Maimon, O. Young, D. Goren-Bar, S. Martins, L. Vaszar, Y. Liel, A. Yarkoni, M. K. Goldstein, A. Leibowitz, and TMarom, "The Importance of Creating an Ontology-Specific Consensus Before a Markup-Based Specification of Clinical Guidelines," Proc biennial European Conference on Artif Intel, Riva del Garda, Italy, 2006.

[29] Ruby Community, "Ruby Language Home Page" http://www.ruby-lang.org/en/

[30] W. E. Grosso, H. Eriksson, R. Fergerson, J. H. Gennari, S. W. Tu, and M. A. Musen, "Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000)," 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop., Banff, Canada, 1999.

[31] S. Siegel, Non-Parametric Statistics for the Behavioral Sciences. New York: McGraw-Hill, 1956.

[32] L.C. Briand, Y. Labiche, M. DiPenta, and H.D. Yan-Bondoc, "An Experimental Investigation of Formality in UML-Based Development," *IEEE Trans. Software Eng.*, vol. 31, no. 10, pp. 833-849, 2005.

[33] C.J. Hou, M.A. Musen, and N.F. Noy, "EZPAL: Environment for composing constraint axioms by instantiating templates," *Intl. J. Human-Computer Studies*, vol. 62, no. 5, pp. 578–596, 2005.

Dizza Beimel has been a Lecturer in the Department of Business Management of the Ruppin Academic Center, Israel, since 2005. Dizza received her BSc degree in Electrical Engineering from the Technion – Israel Institute of Technology, and her MSc in Information Systems Engineering from the Technion's Faculty of Industrial Engineering. She received her Ph.D. from the same faculty in 2008. The title of her thesis was "Privacy-preservation mechanisms for accessing Electronic Health Record Data."

Mor Peleg has been a Senior Lecturer in the Department of Management Information Systems at the University of Haifa, Israel, since 2003. Mor received her BSc and MSc in Molecular Biology from the Technion – Israel Institute of Technology, and a Ph.D. in Information Systems Engineering from the Faculty of Industrial Engineering at the Technion in 1999. She completed a 4-year post-doc fellowship at Stanford Medical Informatics, where she also was a visiting assistant professor during a sabbatical in 2007-9. In 2005 she was awarded the New Investigator Award by the American Medical Informatics Association (AMIA). Mor's research concerns the development of methodologies and software tools for representing and analyzing complex biolmedical systems. Her research has appeared in such publications as JAMIA, JBI, Methods of Information in Medicine, Intl J of Medical Informatics, Bioinformatics, IEEE T on Software Engineering, and Proc. IEEE.

APPENDIX A: SCENARIOS USED IN THE EXPERIMENT

Table 11 shows the three scenarios used in assignment 1 of the experiment. The three scenarios were shown to the

students as instances of the Context or SitBAC ontologies in Protégé. The students were asked to textually phrase the scenarios. The correct answers are shown in Table 11.

Tables 12 and 13 summarize the simple-comprehension and complex-comprehension tasks incorporated in assignment 1. The correlated analysis results appear in Tables 2 and 3.

In assignments 2 and 3, the students received the descriptions presented in Tables 14 and 15, Scenarios 4.1 and 4.2. They had to create six corresponding instances using Protégé.

TABLE 11: SCENARIOS DESCRIPTION FOR ASSIGNMENT 1

Scenario 2.1	Physician is allowed to order prescription if the patient is inpatient and the location of access is in the clinic area.
Scenario 2.2	Nutritionist is allowed to view diagnosis if the patient is inpatient, the location of access is within the emergency domain, and the time of access is 19:00-7:00.
Scenario 2.3	Audit Physician is allowed to view prescription if the patient is inpatient and the patient insurance is within the audit physician employer.

TABLE 12: SIMPLE-COMPREHENSION SUB-TASKS FOR TASK 1

Scenario	Identified DR Role	Identified Response	Identified Task Action	Identified Task EHR Section	Total
2.1	1	1	1	1	4
2.2	1	1	1	1	4
2.3	1	1	1	1	4

TABLE 13: COMPLEX-COMPREHENSION TASKS FOR ASSIGN.1

Scenario	Identified DR Location of Access	Identified DR Time of Access	Identified Patient's Status	Identified Patient – DR Relation	Total
2.1			1		1
2.2	1	1	1		3
2.3			1	1	2
Total:					**6**

TABLE 14: SCENARIOS DESCRIPTION FOR ASSIGNMENT 2

Scenario 3.1	Clinical researcher is allowed to view prescription and diagnosis of inpatient. However, he is not allowed to view identification section of inpatient.
Scenario 3.2	Audit physician is allowed to view diagnosis of her inpatient who is not insured by the audit physician's employer, if the audit physician is in the emergency room.
Scenario 3.3	Nurse is allowed to view diagnosis of inpatients, but is not allowed to view diagnosis of outpatients while the nurse is visiting in the clinics.

TABLE 15: SCENARIOS DESCRIPTION FOR ASSIGNMENT 3

Scenario 4.1	Registered nurse is allowed to view diagnosis of an inpatient if he is located in the Pediatrics unit and her workplace is the Pediatrics unit.
Scenario 4.2	Paramedic is allowed to view tests of inpatients during his shift.
Scenario 4.3	Family Doctor is allowed to view the whole EHR of his patient, who has to be insured by the family doctor's employer, while the doctor is in the primary care clinic

Table 16 summarizes the simple-existing tasks that we identified in assignments 2 and 3. The corresponding analysis results appear in Table 4. Table 17 summarizes the complex-existing tasks and Table 18 the simple-new tasks, incorporated in tasks 2 and 3. The corresponding analysis results appear in tables 5 and 6. Table 19 summarizes the complex-new tasks for assignments 2 and 3. The corresponding analysis results appear in Table 7.

TABLE 16: SIMPLE-EXISTING TASKS FOR ASSIGNMENT 2 AND 3

Scenario	Express DR Role	Express Response	Express Task Action	Express Task EHR Section	Total
3.1		1	1	2	4
3.2	1	1	1	1	4
3.3		2	1	1	Bonus- not counted
4.1		1	1	1	3
4.2		1	1		2
4.3		1	1		2
Total:					**15**

TABLE 17: COMPLEX-EXISTING TASKS FOR ASSIGNMENT 2 & 3

Scenario	Express DR location	Express DR Role	Express Patient-DR relation	Total
3.1		**1**		**1**
3.2	**1**	**1**		**2**
3.3	**1**	**2**		Bonus- not counted
4.1		**1**		**1**
4.2		**1**		**1**
4.3			**1**	**1**
Total				**6**

TABLE 18: SIMPLE-NEW TASKS FOR ASSIGNMENT 2 AND 3

Scenario	Express DR Role	Express Task EHR	Total
3.1	**1**	**Disallowing access possible only in SitBAC**	**1**
3.2			
3.3	**1**		Bonus- not counted
4.1	**1**		**1**
4.2	**1**	**1**	**2**
4.3	**1**		**1**
Total			**5**

TABLE 19: COMPLEX-NEW TASKS FOR ASSIGNMENT 2 AND 3

Scenario	Express DR Location of Access	Express DR during Shift	Express DR Workplace	Express Patient's Location	Express Relation between Patient and DR	Total
3.1						
3.2					**1**	**1**
3.3						
4.1			**1**	**1**		**2**
4.2		**1**				**1**
4.3	**1**				**1**	**2**
Total						**6**