Building an Incident Management Body of Knowledge

David A. Mundie
CERT Program
Software Engineering Institute
Pittsburgh, PA, USA
dmundie@cert.org

Robin Ruefle
CERT Program
Software Engineering Institute
Pittsburgh, PA, USA
rmr@cert.org

***Abstract*—The CERT Incident Management Body of Knowledge (CIMBOK) was built using a systematic process that starts with a controlled vocabulary and evolves through taxonomies, static ontologies, dynamic ontologies, intentional ontologies, and metamodels. The CIMBOK builds on 10 previous standards for incident management. This paper describes the components of the CIMBOK and how they were constructed.**

Keywords: body of knowledge; incident response; incident management; taxonomy; static ontology; process model; competency framework.

I. INTRODUCTION

Professional communities have created and used bodies of knowledge (BOKs) to consolidate their discipline, standardize practices, improve processes, and warehouse community knowledge. Formal BOKs have been used across disciplines as varied as medical practice management [1], computer usability [2], personal software process [3], Standard CMMI Appraisal Method for Process Improvement (SCAMPI) [4], software engineering [5], project management [6], and IT security [7].

Inspired by these previous efforts, the CERT® Incident Management Body of Knowledge (CIMBOK) has the following goals:

- Allow the incident management profession to define itself. Even in the United States, incident management is still evolving as a discipline, and in other parts of the world it is not well recognized. A professional community gains a strong sense of definition from its own BOK.
- Enable incident management to be standardized at all levels, including vocabulary, competencies, and process models.
- Facilitate the creation of collective, expandable repositories for knowledge about incident management.
- Provide guidance for developing curricula, training requirements, job competency descriptions, and certification programs for incident management.
- Enable benchmarking, gap analysis, and process improvement of incident management within organizations.

The CERT Program, part of Carnegie Mellon's® Software Engineering Institute (SEI), developed the CIMBOK by analyzing 10 previous standards for incident management:

1. The Department of Homeland Security's Information Technology (IT) Security Essential Body of Knowledge Essential Body of Knowledge (EBK). This ontology includes a vocabulary of IT security terms, defines a Competency Lifecycle Framework CLF-style static ontology of competency areas, and maps competency clusters to job roles [7].
2. The CERT® Resilience Management Model (CERT®-RMM). CERT-RMM provides a process-oriented dynamic ontology that includes five practices and 14 subpractices [8].
3. DoD 8570.01-M: Information Assurance Workforce Improvement Program. This CLF-style static ontology provides 12 computer network defense and incident response competencies [9].
4. Incident Management Capability Metrics. This is another CLF-style static ontology [10].
5. CJCSI Directive 6510: Information Assurance and Support to Computer Network Defense. This is a dynamic ontology of the incident handling process, divided into six phases and 34 subphases [11].
6. ITIL V3 Foundation Handbook. This is a dynamic ontology of the incident management process, consisting of six activities [12].
7. ISO 27002. This static ontology is based on the notion of control [13].
8. The CSIH Certification job task analysis surveys. [14]. These cover 99 incident management tasks.
9. NIST's Computer Security Incident Handling guide [15]. This uses a six-phase process model.

® CERT® is a registered mark owned by Carnegie Mellon University.

® Carnegie Mellon is a registered mark owned by Carnegie Mellon University.

10. CERT's incident management process [16]. This dynamic ontology is based on a four-phase process model with 14 subphases

We synthesized a metamodel that captures the commonalities among those 10 standards. We then used that model as input to a formal process for developing BOKs that was derived from an investigation of existing BOKs and from a review of knowledge representation theory.

II. Building an Incident Management Body of Knowledge

BOKs are a relatively recent development in information modeling, but they draw on a rich heritage from other models. Unfortunately the Body of Knowledge BOK remains to be written, so it can be difficult to understand how BOKs relate to other knowledge representations. Nonetheless we believe those relationships are essential to the disciplined creation of BOKs.

This section, based on a review of the literature, provides a coherent overview of the logical evolution of a BOK in four phases:

1. *Controlled vocabulary*. A collection of preferred terms that are used to more precisely retrieve content, categorize content, build labeling systems, create style guides, and design database schemata. (Adapted from [17].)
2. *Taxonomy*. A set of hierarchically related terms in a controlled vocabulary. (Adapted from [17].)
3. *Ontology*. A set of statements about a knowledge domain consisting of terms from a controlled vocabulary and the relationships among them. We follow Jurisica et al. [18] in distinguishing several subtypes of ontologies:
    a. *Static ontology*. An ontology that "describes static aspects of the world, i.e., what things exist, their attributes and relationships."
    b. *Dynamic ontology*. An ontology that "describes the changing aspects of the world in terms of states, state transitions, and processes."
    c. *Intentional ontology*. An ontology that "encompasses the world of things agents believe in, want, prove, disprove, or argue about." We extend this definition to include the knowledge and skills those agents possess.
4. *Metamodel*. An ontology template whose parameters can be set to generate ontologies. The metamodel seeks to discover underlying similarities between the BOK being developed and other, related BOKs.

The term "body of knowledge" is used to refer to many different combinations of these formal models It is frequently used to mean simply a static ontology for a particular discipline, although it can also refer to an intentional ontology, a process model, or simply a controlled vocabulary.

Once a BOK is defined, it can become a convenient place to store a wide variety of miscellaneous information such as bibliographies, tips, and anecdotes. The often hodge-podge nature of BOKs can obscure their underlying formal model This formal model should not be ignored, but we believe that its obfuscation indicates the success of BOKs, not their failure.

A. Controlled Vocabularies

A BOK starts with a simple list of terms. The CIMBOK includes such as

- incident
- event
- zero-day exploit
- service

The next step is to define the terms, adhering to two best practices of lexicography to make the vocabulary as useful as possible. First, definitions should avoid ambiguity by discriminating among homographs such as the verb "compromise" and the noun "compromise." Second, the vocabulary should support a standardized, canonical usage by avoiding synonyms. For example, the term "buffer overflow" is preferred over "buffer overrun." Some examples definitions from the CIMBOK include the following:

- incident: A violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices.
- event: Any observable occurrence in a network or system.
- zero-day exploit: An exploit that takes advantage of a security vulnerability on the same day that the vulnerability becomes generally known.
- service: A set of activities that an organization carries out in the performance of a duty or in the production of a product.

When practitioners commit to using these terms, with or without formal enforcement, and when the list is under some form of configuration management, the result is called a *controlled vocabulary*. (For more information about controlled vocabularies, David Riecks maintains a useful repository [19].)

Good lexicographic practice is *descriptive* rather than *prescriptive*, which frequently requires documenting multiple senses for the same word and capturing, when appropriate, usage information for the various senses [20].

Controlled vocabularies are often augmented with additional nonstandard terms and mappings into the equivalent standard terminology, for example

- monitoring report: See *audit log*.
- incomplete data: See *bad data*.

This is known as a thesaurus, and the subject heading catalogs used by libraries are well-known examples.

The CIMBOK controlled vocabulary contains a 2,000-word general information security dictionary, a 700-word insider threat dictionary, a 200-word malicious code dictionary, and dictionaries of terms used by the CERT-RMM and the Capability Maturity Model (CMM). These dictionaries are implemented using the DICT standard (RFC

2229 [21]) from the Internet Engineering Task Force (IETF)..

B. Taxonomies

The terms in a controlled vocabulary often have an internal structure that can be modeled as a tree. This classification reflects the "is-a" relationships in the vocabulary and captures the hierarchy of levels of abstraction in the domain. To illustrate using terms from the Longstaff and Howard taxonomy ("attack action" [22]):

- attack action:
  - probe…
  - scan…
  - flood…
  - authenticate…
  - bypass…
  - …

Here a probe is an attack action, a scan is an attack action, a flood is an attack action, and so on.

If the classification in a controlled vocabulary reflects the way the entities in question evolved, so that the relation is not just "is-a" but also "is-a-child-of," then the classification is a taxonomy. (For more information, a special interest group of the American Society for Indexing collects useful information about the use of taxonomies in knowledge representation [23].)

A good example of a taxonomy in incident management is the family tree of the Bagle virus. A small portion of that tree is given here in textual form and a larger portion in graphic form in Fig. 1:

W32/Bagle.n@MM

W32/Bagle.z@MM

W32/Bagle.af@MM

W32/Bagle.ag@MM

W32/Bagle.ad@MM

W32/Bagle.p@MM

W32/Beagooz

W32/Bagle.cb@MM

In this taxonomy, W32/Beagooz and W32/Bagle.cb@MM are two "species" of the "genus" W32/Bagle.p@MM [24]. The presence of parent-child relationships in this classification makes it a taxonomy.

The distinction between a classification and a taxonomy is not always observed in practice. Popular parlance often uses "taxonomy" to refer to any hierarchical classification, whether it represents an evolutionary process or not. Further, it would seem logical to allow metaphorical taxonomies, such as the evolution of different network sniffing tools, but it is not clear whether this is completely accepted by the community.

A classification of the terms in the CIMBOK vocabulary is being developed. The mind map in the figure below illustrates this classification for insider threat terms.

Apart from family trees of malware, we have not found many uses for taxonomies in incident management.

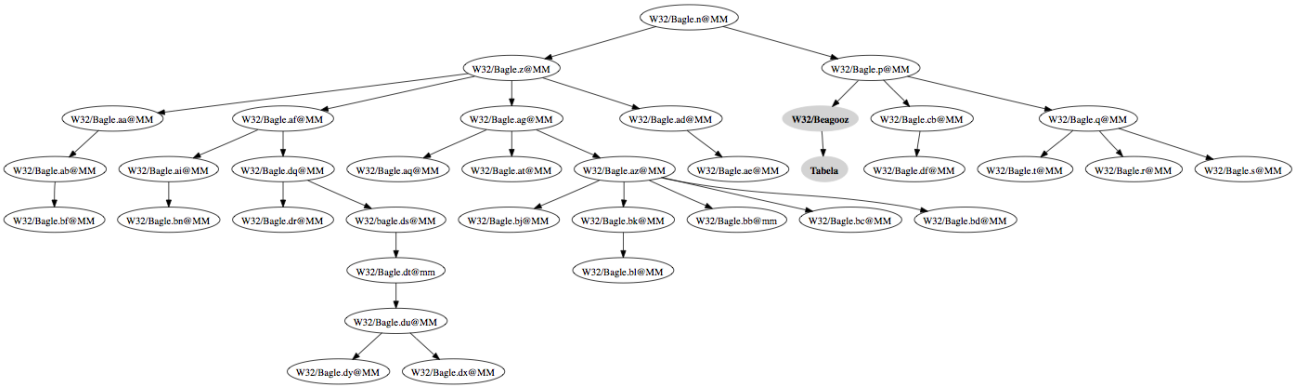

Figure 1. Taxonomy of the Bagle virus.

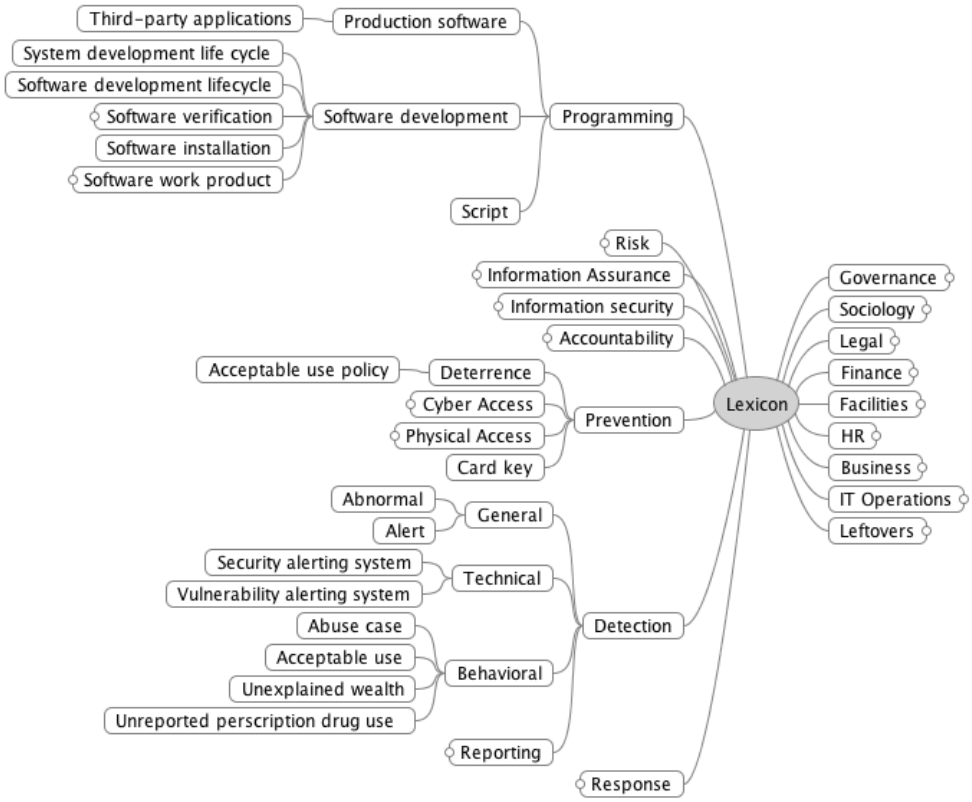

Figure 2. A preliminary categorization of the insider threat terms in the CIMBOK dictionary.

C. *Static Ontologies*

Once the standardized terms have been organized into categories, the next step is to make statements of facts within the knowledge domain of interest. These statements take the form of terms connected by the relationships between them, as seen in the following examples:

- Attackers exploit vulnerabilities.
- Smurf attacks utilize the ICMP protocol.

The first example asserts the "exploit" relationship between "attackers" and "vulnerabilities," and the second asserts the "utilize" relationship between "Smurf attacks" and "the ICMP protocol."

Mylopolos [25] calls this a static ontology, and it is a common component of a BOK. It is essentially the same as the entity-relationship diagrams used to design relational databases.

When a static ontology is represented in a formal notation, it is referred to as a "formal ontology." For example, the following might represent the statements above in a formal ontology:

- `Exploit(attackers, vulnerabilities)`
- `Utilize(Smurf attacks, ICMP)`

Here the sans-serif font and the function-form notation suggest that these two statements can be processed by some software that manipulates formal ontologies.

The Web Ontology Language (OWL) [26] and the DARPA Agent Modeling Language (DAML) [27] are formal languages that were designed specifically to allow machine-based reasoning about ontologies in the context of the World Wide Web.

An essential component of the CIMBOK is a static ontology of 340 incident management activities drawn from the 10 preexisting process models described in the Introduction. Each activity is classified according to where it occurs in the incident management lifecycle, the skills it requires, its relevant knowledge domains, and the standard in which it is described. A faceted interface [28, 29] that allows easy browsing of the static ontology has also been developed.

D. *Dynamic Ontologies*

Entity-relationship models are inherently atemporal; they abstract away from the details of how the model actually evolves. To capture the temporal dimension, relationships must be aggregated into sequences of steps to form a *dynamic ontology*. For example, one possible process for analyzing root causes, borrowed from [30], might be:

The Root Cause Analysis Process

1. Collect data
2. Construct causal factor chart
3. Identify the root cause

4. Generate recommendations and implementation plans

The process improvement community more commonly knows such dynamic ontologies as *process models*. They are still quite general, but they capture the temporal relationships among their elements.

Frequently a process needs to be captured in more detail and have its process attributes documented. This is what is referred to as a *process definition* [31]. In our root cause analysis domain, we might want to capture some of the details of the data collection process as well as the entry and exit conditions:

The Root Cause Analysis Process

Entry conditions: an incident report has been received and triaged

Steps:

a. Collect as much data as possible, striving for a complete and accurate understanding the event and its causal factors
b. Construct a causal factor chart that contains logic tests describing the events leading up to the occurrence
c. Use a decision diagram (the Root Cause Map) to identify the underlying reason for each causal factor
d. Use the root cause analysis to generate recommendations for preventing the recurrence of the incident

Exit conditions: the root cause has been identified and recommendations made (but not implemented)

Only a small percentage of the processes in the CIMBOK have been captured as process definitions. A challenge to developing such definitions is the nondeterministic nature of many incident management activities, such as artifact analysis, which do not lend themselves to linear process steps. We are currently investigating the objective-based approach used by Beebe and Clark [32] as a possible solution.

E. Intentional Ontologies

Arguably the most important entity in a BOK's static ontology is the practitioner—the individual who will be participating in the discipline of interest. This leads many BOK developers to focus on the competencies required of practitioners. For example, the SCAMPI Lead Appraiser Body of Knowledge uses the SEI's Competency Lifecycle Framework [33].

Competency frameworks can be viewed in two ways. The first way is as an abstraction of a set of process definitions, which mostly abstract away the details and the temporal dimension of the processes, leaving the competencies and the subcompetencies (skills) that are required to carry out the process. The second way is as a static ontology whose entities are the actors in a process and the attributes they possess—what Jurisica calls an *intentional ontology* [18]. The competency framework in our root cause analysis example might be the following:

Competency 1.1 Performing Root Cause Analysis

Associated Skills

- Gathering and assessing data
- Managing collected data
- Analyzing complex causal chains
- Solving problems based on an analytical framework

This competency framework might alternatively be represented as an intentional ontology:

```
Incident management staff → Can-collect
→ incident data


Incident management staff → Can-analyze
→ causal chains in incident data

Incident management staff → Can-solve-
problems → to break causal chains
```

The CIMBOK incorporates competency information for each of the tasks it encompasses, making it possible to automatically generate traditional competency matrices, as illustrated by the figure below.

	Problem Solving	Analytical Thinking	Project Planning	Communicating	Integrating Information	Articulating Information	Operating Computers
Preparing for incidents		Perform risk assessments	Develop an IR plan	Develop trusted relationships			
		Apply risk assessments		Train constituents			
				Train staff			
Protecting against incidents	Improve Defenses						Monitor systems
Detecting incidents		Detect and report events					
Analyzing incidents		Analyze incident				Document incident analysis	
		Triage incident					
Responding to incidents		Remove cause of incident			Collect and preserve evidence	Track and document incidents	Maintain a chain of custody
Recovering from incidents							Restore and validate the system
							Close the incident
Learning from incidents	Identify improvement actions	Perform a postmortem review		Communicate incidents			
	Integrate lessons learned						

Figure 3. An incident management competency matrix.

F. Metamodels

The last step in developing a BOK is to abstract away from its specificity to capture the similarities between it and other related BOKs. The result is a metamodel, which is the combination of abstract ontologies and domain models. Under the guidance of the Object Management Group, this approach has achieved considerable success in software development, where a variety of software programs can be seen as instantiations of the same underlying metamodel [34].

In our incident management example, we might define a generalized "incident" ontology containing relationships such as

- First responder, performs initial analysis on, incident
- Incident handler, researches, remediation methods

and combine it with a domain model that specifies the substitutions that instantiate the ontology in different domains such as firefighting and helpdesk support:

Firefighting

first responder = fireman
incident = fire
initial analysis = how many fire trucks will be needed
remediation method = stricter building codes

Helpdesk

first responder = helpdesk staff
incident = information security incident
initial analysis = type of incident
remediation method = security configuration management

A metamodel's degree of abstraction entails an engineering tradeoff that balances the economy of reuse against the costs of reinstantiating the metamodel. For example, a metamodel for food preparation that covers every culinary technique is unlikely to provide much benefit. On the other hand, a metamodel for preparing omelets will almost certainly provide much greater benefits and be much more efficient than developing individual models for every type of omelet.

The incident management community makes frequent use of informal metamodels, for example, comparing incident response teams to emergency medical teams or developing families of related incident management capabilities. However, the CIMBOK has no formal metamodel component at this time.

REFERENCES

[1] Body of Knowledge for Medical Practice Management. Englewood, CO: Medical Group Management Association, 2006.

[2] Nigel Bevan (ed.), Usability Body of Knowledge. Bloomingdale, IL: Usability Professionals' Association, 2005, http://www.usabilitybok.org.

[3] Marsh Pomeroy-Huff, Julia Mullaney, Robert Cannon, and Mark Sebern, Personal Software Process (PSP) Body of Knowledge, Version 1.0 (CMU/SEI-2005-SR-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005, http://www.sei.cmu.edu/library/abstracts/reports/05sr003.cfm

[4] Steve Masters, Sandra Behrens, Judah Mogilensky, and Ryan Charles. SCAMPI Lead Appraiser Body of Knowledge (SLA BOK) (CMU/SEI-2007-TR-019). Pittsburgh, PA: Software

Engineering Institute, Carnegie Mellon University, 2007, http://www.sei.cmu.edu/library/abstracts/reports/07tr019.cfm.

[5] Alain Abran and James W. Moore (eds.), Guide to the Software Engineering Body of Knowledge. Los Alamitos, CA: IEEE Computer Society Press, 2004.

[6] Project Management Institute, A Guide to the Project Management Body of Knowledge, 4th ed. Newton Square, PA: Project Management Institute, 2008.

[7] Department of Homeland Security, Office of Cybersecurity and Communications, National Cyber Security Division. Information Technology (IT) Security Essential Body of Knowledge (EBK): A Competency and Functional Framework for IT Security Workforce Development. Washington DC: DHS, 2008, http://www.us-cert.gov/ITSecurityEBK/EBK2008.pdf .

[8] Richard A. Caralli, Julia H. Allen, and David W. White, CERT® Resilience Management Model Version 1.1. Upper Saddle River, NJ: Addison-Wesley, 2011.

[9] Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer. DoD 8570.01-M: Information Assurance Workforce Improvement Program. Washington, DC: 2005.

[10] Audrey Dorofee, Georgia Kilcrece, Robin Ruefle, and Mark Zajicek, Incident Management Capability Metrics Version 0.1 (CMU/SEI-2007-TR-008). Pittsburgh, PA:Software Engineering Institute, Carnegie Mellon University, 2007.

[11] Chairman of the Joint Chiefs of Staff, Information Assurance (IA) and Support to Computer Network Defense (CND). Washington DC: Chairman of the Joint Chiefs of Staff Instruction, 2007.

[12] ITIL V3 Foundation Handbook. London: The Stationery Office, 2009.

[13] Code of Practice for Information Security Management.. ISO/IEC 27002:2005.

[14] Computer Security Incident Handler Project: Verification and Alignment Surveys Combined Results. Ohio State University Center on Education and Training for Employment. Unpublished.

[15] Karen Scarfone, Tim Grance, and Kelly Masone. Computer Security Incident Handling Guide. Gaithersburg, MD: NIST Special Publication 800-61, 2008.

[16] Chris Alberts, Audrey Dorofee, Georgia Killcrece, Robin Ruefle, and Mark Zajicek, Defining Incident Management Processes for CSIRTs (CMU/SEI-2004-TR-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2007.

[17] Victor Lombardi, "Metadata glossary," Noise Between Stations, 2012. http://noisebetweenstations.com/personal/essays.

[18] I. Jurisica et al., "Using ontologies for knowledge management: An information systems perspective," Proc. ASIS Annual Mtg., vol. 36, pp. 482-496, 1999.

[19] David Riecks, "Controlled Vocabulary: One Thing Leads to Another," 2012. http://www.controlledvocabulary.com.

[20] B.T.S. Atkins and Michael Rundell, The Oxford Guide to Practical Lexicography. New York: Oxford University Press, 2008.

[21] R. Faith and B. Martin, "A Dictionary Server Protocol". IETF Network Working Group, Request for Comments 2229, 1997.

[22] John D. Howard and Thomas A. Longstaff, "A common language for computer security incidents," Computer, 1998.

[23] ASI 2012

[24] A. Gupta et al., "An Empirical Study of Malware Evolution," COMSNETS, 2009.

[25] John Mylopoulos et al., "Using ontologies for knowledge management: An information systems perspective," Knowledge and Information Systems (2004) 6:380-401.

[26] Sean Bechhofer et al. OWL Web Ontology Language Reference. Boston: W3C, 2004.

[27] Defense Advanced Research Projects Agency (DARPA), "About the DAML Language," The DARPA Agent Markup Language Homepage, www.daml.org/about.html.

[28] Complete Information Architecture, Inc., "Faceted classification software tools," Facetmap, 2009, http://www.facetmap.com.

[29] William Denton. "How to make a faceted classification and put it on the web," Miskatonic University Press, William Denton, November 2003, http://www.miskatonic.org/library/facet-web-howto.html.

[30] J. Rooney et al., "Root cause analysis for beginners," Quality Progress, July 2004.

[31] Mike Bandor, "Process and Procedure Definition: A Primer". Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2007.

[32] Nicole Lang Beebe and Jan Guynes Clark, "A hierarchical, objectives-based framework for the digital investigations process," Digital Investigation, vol. 2, no. 2, pp. 146-166, 2005.

[33] Sandi Behrens et al., CMMI-Based Professional Certifications: The Competency Lifecycle Framework (CMU/SEI-2004-SR-013). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004, http://www.sei.cmu.edu/publications/documents/04.reports/04sr013/04sr013.html.

[34] Manfred A. Jeusfeld, Matthias Jarke, and Mylopoulos (eds.), Metamodeling for Method Engineering. Cambridge, MA: The MIT Press, 2009.