Integrating Top-down and Bottom-up Cybersecurity Guidance using XML

Joshua Lubell
National Institute of Standards and Technology

Abstract

This paper describes a markup-based approach for synthesizing disparate information sources and discusses a software implementation of the approach. The implementation makes it easier for people to use two complementary, but differently structured, guidance specifications together: the (top-down) Cybersecurity Framework and the (bottom-up) National Institute of Standards and Technology Special Publication 800-53 security control catalog. An example scenario demonstrates how the software implementation can help a security professional select the appropriate safeguards for restricting unauthorized access to an Industrial Control System. The implementation and example show the benefits of this approach and suggest its potential application to disciplines other than cybersecurity.

Table of Contents

1. Introduction

The Cybersecurity Framework [CSF] and National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53 [SP800-53] are complementary cybersecurity guidance specifications. The Cybersecurity Framework helps practitioners raise awareness within an organization and communicate assessments and objectives to stakeholders. SP 800-53 provides a rigorous methodology for tailoring a comprehensive catalog of security controls to meet an organization's risk management needs. The Cybersecurity Framework facilitates top-down decision-making, whereas NIST SP 800-53 enables a more bottom-up approach to managing cyber-risk.

Because the Cybersecurity Framework and NIST SP 800-53 are complementary, using the two together can provide a greater benefit than using either alone. But combining the top-down, mission-focused guidance in the Cybersecurity Framework with the bottom-up risk management guidance in NIST SP 800-53 is a challenge. Markup technologies can synthesize the security guidance from the Cybersecurity Framework and NIST SP 800-53 into a coherent whole.

This paper presents research demonstrating that software implemented entirely in the Extensible Markup Language (XML) [XML] can effectively make it easier for security professionals to use the Cybersecurity

Framework and NIST SP 800-53 together. The research also suggests that the approach presented can be successful in solving the more general problem of developing a user interface (UI) to integrate and synthesize information from disparate sources, provided that the quantity of information and number of sources are small enough to not overwhelm limited computational or software development resources. In other words, this approach is intended to enable a developer whose day job does not primarily involve coding to write platform-independent software that is easy and inexpensive to deploy.

The rest of this paper proceeds as follows. Section 2 provides an overview of NIST SP 800-53 and the Cybersecurity Framework. Section 3 presents the technical approach: first in general terms applicable to any scenario involving integration of disparate guidance sources, and then as applied to the implementation discussed in Section 4. Section 4 introduces Baseline Tailor, a software application, implemented using the approach discussed in Section 3, that makes it easier for people to use the Cybersecurity Framework and NIST SP 800-53 security control catalog together. Section 5 presents an example usage scenario demonstrating how Baseline Tailor can help a security professional select the appropriate safeguards for restricting unauthorized access to an Industrial Control System (ICS). Section 6 summarizes some previous third-party research efforts that influenced this work. Section 7 concludes the paper.

2. Background: NIST SP 800-53 and the Cybersecurity Framework

NIST SP 800-53 provides guidance for selecting and tailoring security controls for information systems. The security controls defined in NIST SP 800-53 should be applied as part of a rigorous risk management process. NIST SP 800-53 organizes its catalog of security controls into eighteen families with each family representing a general security topic. A two-character identifier uniquely identifies the family. Each control has zero or more control enhancements, each of which adds additional functionality to or increases the strength of the control. The catalog specifies three security control baselines: for low, moderate, and high impact information systems. NIST recommends the baselines as starting points for security control selection. For example, an organization looking to select security controls for a low impact system (where the consequences of compromised confidentiality, integrity, and availability of information are low) might begin with the controls in the baseline for the low impact level (or more succinctly, the low baseline) and tailor them as appropriate.

Table 1 shows the low, moderate, and high baselines for the first six controls in the Access Control (AC) family. In most cases, the moderate baseline is a superset of the low baseline, and the high baseline is a superset of the moderate baseline. The numbers in parentheses in the two rightmost columns denote control enhancements, which are declarations of security capability to increase the control's functionality and/or strength. For example, AC-2 (1), which identifies control enhancement (1) of AC-2 (Account Management), states a set of capabilities specific to automated system account management. These capabilities enhance the more general capabilities stated for AC-2, which apply to all types of account management. This paper discusses security control AC-2 in further detail in Section 4, where Figure 6 shows AC-2's XML representation in Baseline Tailor, and in the usage scenario in Section 5.

NIST SP 800-53 also contains guidance for creating and documenting overlays to encourage the sharing of best security practices. An overlay is a set of control customizations applicable to a group of organizations with common security requirements. For example, NIST SP 800-82 (Guide to ICS Security) [SP800-82] specifies an overlay for Industrial Control Systems, which are common in the utility, transportation, chemical, pharmaceutical, process, and durable goods manufacturing industries. An ICS is vulnerable to many of the same security threats that affect traditional information systems, yet has unique needs requiring additional guidance beyond that offered by NIST SP 800-53.

The Cybersecurity Framework provides a way for organizations to describe their current security posture and target state, and to communicate and assess progress toward meeting goals. The Cybersecurity

Framework is organized in a hierarchical fashion, which allows for high-level as well as detailed descriptions of security outcomes. It can facilitate communication not only between different categories of stakeholders but also between different levels of management within an organization, for example, between a chief executive and cybersecurity professionals responsible for implementation. In addition, the Cybersecurity Framework links desired security outcomes to specific NIST SP 800-53 security controls, as well as to sections of other standards, guidelines, and best practices offering guidance on how to achieve desired cybersecurity outcomes. This paper focuses specifically on the links to NIST SP 800-53.

Table 1.

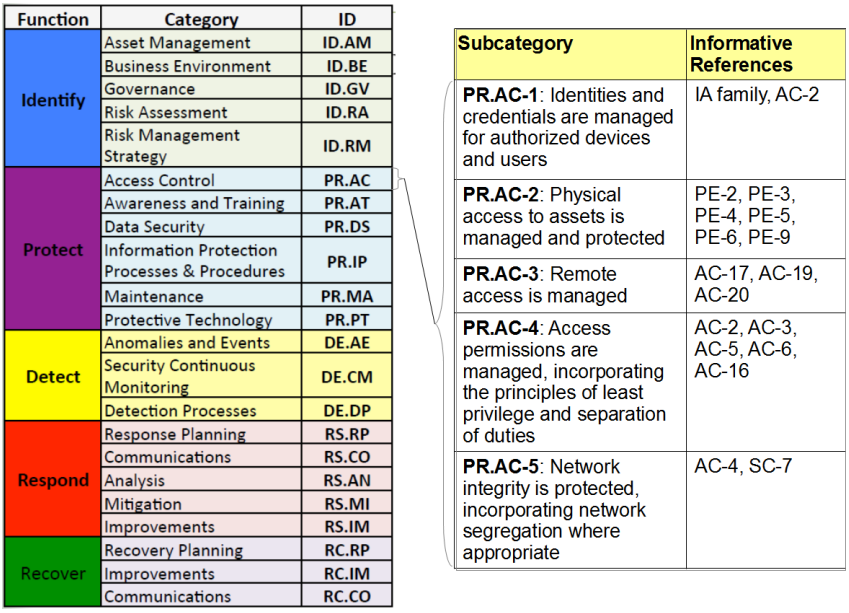
Low, moderate, and high baselines for the first six controls in the Access Control (AC) family.

ID	NAME	LOW	MODERATE	HIGH
AC-1	Access Control Policy and Procedures	AC-1	AC-1	AC-1
AC-2	Account Management	AC-2	AC-2 (1) (2) (3) (4)	AC-2 (1) (2) (3) (4) (5) (11) (12) (13)
AC-3	Access Enforcement	AC-3	AC-3	AC-3
AC-4	Information Flow Enforcement	Not Selected	AC-4	AC-4
AC-5	Separation of Duties	Not Selected	AC-5	AC-5
AC-6	Least Privilege	Not Selected	AC-6 (1) (2) (5) (9) (10)	AC-6 (1) (2) (3) (5) (9) (10)

A major component of the Cybersecurity Framework is the Framework Core, a taxonomy of cybersecurity outcomes common across critical infrastructure sectors. The highest level of the Framework Core consists of five overarching cybersecurity functions: “Identify”, “Protect”, “Detect”, “Respond”, and “Recover”. Each function has a two-character identifier: ID for “Identify”, PR for “Protect”, DE for “Detect”, RS for “Respond”, and RC for “Recover”. Each function is subdivided into categories, which are high-level outcomes. Each category's identifier consists of its function identifier, followed by a period, followed by two more characters such that the category identifier uniquely identifies the category. Each category in turn contains a set of subcategories, which are specific lower-level outcomes that support the category’s higher-level outcome. Subcategories are identified numerically in a manner similar to that of security controls within a control family. Each subcategory has informative references providing guidance for achieving the subcategory’s outcome, including references to NIST SP 800-53 security control definitions. The NIST SP 800-53 informative references are essential for synthesizing the Cybersecurity Framework and NIST SP 800-53 guidance, as will be shown in Section 4.

Figure 1 shows the Framework Core functions and categories, with the “Protect” function's “Access Control” category (PR.AC) expanded to show all five of its subcategories. The Informative References column on the right only shows references to NIST SP 800-53. References to other standards, guidelines, and best practices are excluded because they are out of scope for this paper. As this column shows, the Cybersecurity Framework is less granular than NIST SP 800-53. References are to controls in their entirety, and do not distinguish between control enhancements or baselines.

Figure 1.



Function	Category	ID
Identify	Asset Management	ID.AM
	Business Environment	ID.BE
	Governance	ID.GV
	Risk Assessment	ID.RA
	Risk Management Strategy	ID.RM
Protect	Access Control	PR.AC
	Awareness and Training	PR.AT
	Data Security	PR.DS
	Information Protection Processes & Procedures	PR.IP
	Maintenance	PR.MA
	Protective Technology	PR.PT
Detect	Anomalies and Events	DE.AE
	Security Continuous Monitoring	DE.CM
	Detection Processes	DE.DP
Respond	Response Planning	RS.RP
	Communications	RS.CO
	Analysis	RS.AN
	Mitigation	RS.MI
	Improvements	RS.IM
Recover	Recovery Planning	RC.RP
	Improvements	RC.IM
	Communications	RC.CO

Subcategory	Informative References
PR.AC-1: Identities and credentials are managed for authorized devices and users	IA family, AC-2
PR.AC-2: Physical access to assets is managed and protected	PE-2, PE-3, PE-4, PE-5, PE-6, PE-9
PR.AC-3: Remote access is managed	AC-17, AC-19, AC-20
PR.AC-4: Access permissions are managed, incorporating the principles of least privilege and separation of duties	AC-2, AC-3, AC-5, AC-6, AC-16
PR.AC-5: Network integrity is protected, incorporating network segregation where appropriate	AC-4, SC-7

Cybersecurity Framework Core with expansion of category PR.AC.

A Framework Profile is a subset of the outcomes in the Framework Core representing either an organization's current or target security posture. The Cybersecurity Framework is not prescriptive with respect to how an organization should create a Profile, or how much information a Profile should include beyond an enumeration of the Framework Core subcategories it includes. However, the Cybersecurity Framework suggests that an organization consider basing a Profile on business drivers and an assessment of and tolerance for risk. The Baseline Tailor usage scenario discussed in Section 5 involves use of a Framework Profile to support the selection of NIST SP 800-53 security controls. This scenario specifically illustrates how a Framework Profile focusing on category PR.AC (Access Control) can support selection of security control AC-2 (Account Management).

3. An XML-based Integration Approach

For a general integration approach, applicable for other disciplines besides cybersecurity, consider a generic scenario where multiple information sources need to be combined such that the combined information can be efficiently viewed and manipulated using a common UI. These information sources may or may not be structured XML data. For example, they may be in the form of tables in a document, or as spreadsheets. These information sources can be thought of as Small Arcane Nontrivial Datasets [Lubell2014]. Although not large enough to justify a heavyweight, server-based database application, a Small Arcane Nontrivial Dataset is complex enough to benefit from specialized software for manipulation and access, and important enough to justify the development of such software. Let us further assume a requirement that any results of manipulating the data be presented to the user as structured XML. The following general approach for developing such software that meets the aforementioned requirements uses three XML technologies: XForms, Extensible Stylesheet Language Transformations (XSLT), and the XML Path Language (XPath).

XForms [XForms], an XML application for specifying forms for the Web, is well-suited for implementing UIs for Small Arcane Nontrivial Datasets. XForms adopts the model-view-controller software pattern, making it a good fit for lightweight, data-driven applications. The XForms model consists of a set of instances and a set of bindings. The instances are well-formed XML documents, some static and some dynamic. The bindings define UI constraints, compute dynamic instance data values from other instance

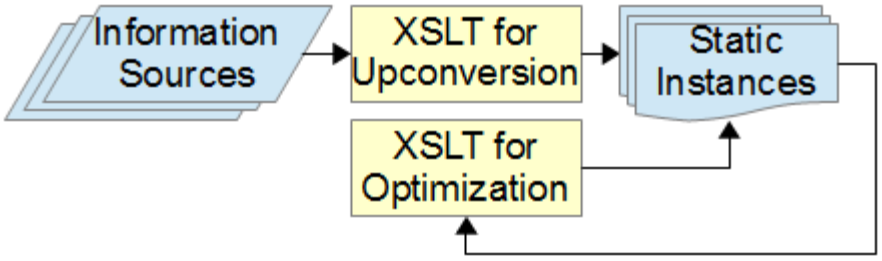
data, and manage the display of UI widgets. Because XForms is an XML language, XForms is a good choice for implementations where data is already available as XML, or when XML output is desired. XForms provides a platform-independent set of UI widgets, enabling the same XForms-valid source code to run in multiple browser environments and on multiple operating systems.

Since XForms requires model instances to be well-formed XML, the original information sources may need to be converted to XML from their native formats. XSLT [XSLT] is particularly well-suited for such a task, even if the source data is non-XML or semi-structured as is the case with Small Arcane Nontrivial Datasets that are spreadsheets or tabular data extracted from documents. If the source is poorly structured, a semi-automated approach combining XSLT with hand-editing may be needed. XSLT is also useful for making flat data hierarchical or vice versa. Additionally, XSLT can be used to create multiple alternatively-structured XForms instances in order to speed up UI operations (at the expense of memory requirements — a space-time tradeoff).

XForms and XSLT both depend on XPath [XPath]. XForms uses XPath for bindings within the model as well as for specifying interactions between the UI widgets and the model. XSLT uses the XPath data model and XPath's library of functions and operators.

Figure 2 shows a generic pipeline for producing static XForms model instances from native information sources. The pipeline uses XSLT to up-convert an unstructured or semi-structured information source into a well-formed, well-structured instance. XSLT is also used to create additional static instances optimized for specific UI operations.

Figure 2.



Generic XML transformation pipeline to produce XForms static model instances.

In the event that the native information source is too poorly structured to support transformation without human intervention, the following semi-automated procedure for extracting tabular data from a semi-structured documentary source can be used:

1. If the document is not in an Office Open XML [ISO29500] Spreadsheet (`.xlsx`) format, save it in `.xlsx` form (see Disclaimer).

2. Determine how the information should be represented as structured XML. This is primarily a data modeling exercise.

3. Open up the result in a spreadsheet authoring software application and, using copy/paste, partition the file into separate Office Open XML Spreadsheet documents such that each document contains a simple tabular spreadsheet with no split cells or cells spanning multiple rows or columns.

4. For each tabular spreadsheet document, create a mapping from columns to XML elements and, using the map, convert the spreadsheet to structured XML.

5. Using XSLT, combine the XML documents as desired, and up-convert ill-structured data within cells as required.

4. Baseline Tailor Overview and Implementation

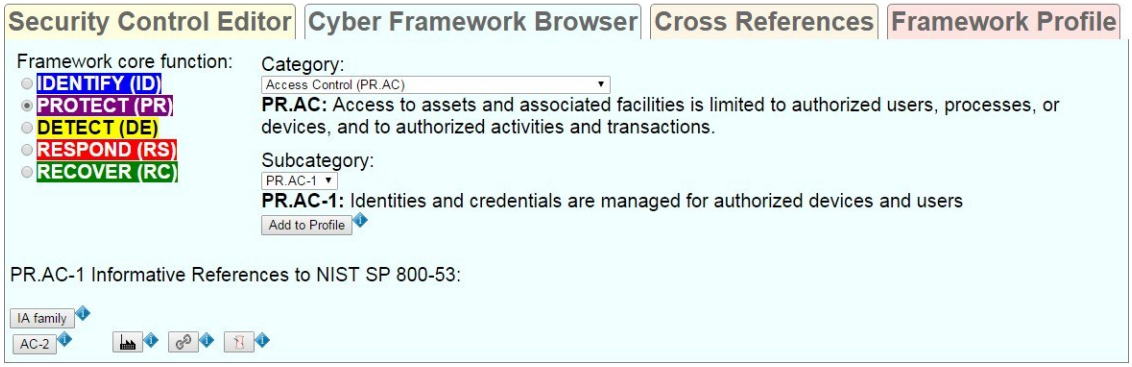
The generic recipe described in the previous section was applied to develop Baseline Tailor, a freely available and open source software tool specifically for users of the Cybersecurity Framework and NIST SP 800-53 security controls. The Baseline Tailor User Guide [Lubell2016] describes this software, and multiple usage scenarios, in detail. [Lubell2015] provides some implementation details, not discussed in this paper, that are specific to Baseline Tailor's UI for tailoring security controls. Section 5 describes a specific Baseline Tailor usage scenario: synthesizing into a coherent whole the security guidance from NIST SP 800-53, the Cybersecurity Framework, and the NIST SP 800-82 ICS overlay. Without Baseline Tailor, an individual wishing to use these specifications together would have to deal with three separate information sources, each organized differently. Baseline Tailor's UI makes it easier to use the specifications together. Additionally, Baseline Tailor provides new information derived through integrating the disparate information sources – information not obvious from studying each specification in isolation.

A Baseline Tailor user utilizes the Cybersecurity Framework to determine an organization's desired security posture, and then tailors an appropriate subset of SP 800-53 security controls needed to make that desire a reality. The Baseline Tailor UI lets users see how Cybersecurity Framework core functions, outcomes and SP 800-53 security controls all relate to one another. It also automatically enforces SP 800-53 tailoring rules. Additionally, the UI produces output in XML so results can be fed directly to other software tools to generate reports, share requirements, or establish assurance. [Lubell2016] discusses Baseline Tailor's XML format for tailored controls, UI support for tailoring controls, and automated SP 800-53 enforcement in detail.

The Baseline Tailor UI, shown in Figure 3, has four tabs:

- A Security Control Editor tab for navigating the NIST SP 800-53 security control catalog and tailoring controls.
- A Cyber Framework Browser tab for navigating the Framework Core and modifying a Framework Profile, the active tab in Figure 3.
- A Cross References tab showing all references from the Framework Core to a particular security control.
- A Framework Profile tab for modifying a Framework Profile and showing the currently-selected subset of Framework Core outcomes.

Figure 3.



Cyber Framework Browser tab.

Figure 4 shows the transformation pipeline used to produce the Baseline Tailor XForms static model instances. This pipeline is a specialization of the pipeline in Figure 2. The pipeline transformed the following native information sources, enclosed by a coarsely dashed border in Figure 4:

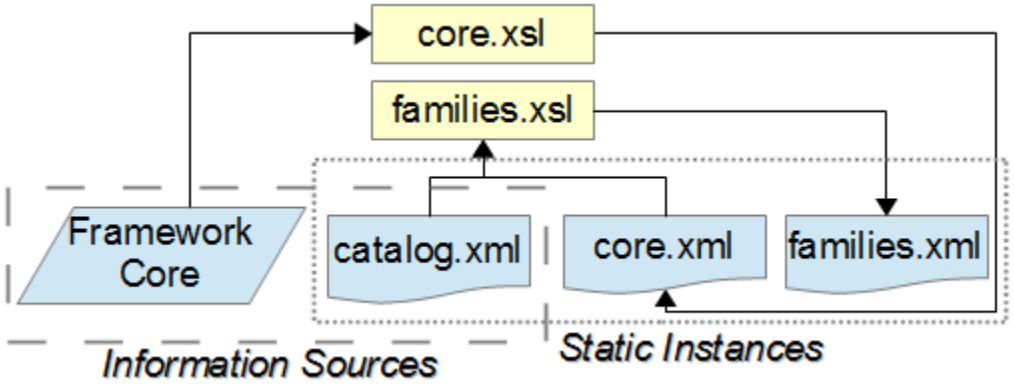
- A tag-delimited tabular representation of the Framework Core, obtained from a Filemaker Pro runtime database (see Disclaimer) available from the Cybersecurity Framework website [CSFTool].

- `catalog.xml`: the structured XML representation of the NIST SP 800-53 security control catalog available from the NIST SP 800-53 database [NVD]. Since the security catalog's native format is structured XML, it is usable as-is as an XForms model instance.[1] Therefore, Figure 4 shows `catalog.xml` as enclosed within both the coarsely-dashed border surrounding the information sources and the finely-dashed border surrounding the XForms static instances. Baseline Tailor uses the data in catalog.xml to generate the portion of the UI in the Security Control Editor tab for tailoring a security control and its control enhancements. Figure 11 shows this portion of the UI when a user has selected security control AC-2 for tailoring.

The XSLT stylesheet `core.xsl` up-converted the semi-structured Framework Core data into a hierarchically structured XForms static instance `core.xml`. Baseline Tailor uses the data in `core.xml` to generate the "Framework core function" radio buttons, "Category" and "Subcategory" drop-down lists, and "Informative References" buttons shown in Figure 3.

The XSLT stylesheet `families.xsl` generated a static instance `families.xml` using the data in `catalog.xml` and `core.xml`. `families.xml` is optimized to facilitate retrieval of security controls belonging to a family, and adds for each security control the identifiers from `core.xml` identifying the Framework Core subcategories that reference the control. The subcategory identifiers are vital to Baseline Tailor for integrating the Cybersecurity Framework and NIST SP 800-53 guidance. Baseline Tailor uses the subcategory information in `families.xml` to generate the information shown in the Cross References tab. Figure 12 shows the Cross References tab after a user has requested the cross references for security control AC-2.

Figure 4.



XML transformation pipeline used to produce Baseline Tailor XForms static model instances.

The XML shown in Figure 5 and Figure 6 illustrates how the Baseline Tailor XForms model represents security controls, subcategories, and their inter-relationships. Figure 5 shows how `core.xml` represents the category PR.AC (shown earlier as a table in Figure 1). Each `category` element has an `id` attribute and contains `subcategory` elements representing the category's subcategories. To reduce Figure 5's verbosity, only the subcategories with informative references to security control AC-2 — PR.AC-1 and PR.AC-4 — are shown in full detail.

[1]Actually, Baseline Tailor does not use the original catalog XML as-is. The original source contains detailed prose text statements from the NIST SP 800-53 Revision 4 document describing each security control in the catalog. Baseline Tailor's UI does not need these descriptions, so they were stripped from Baseline Tailor's `catalog.xml` model instance for efficiency reasons. However, it is fair to say that Baseline Tailor *could* — at least in theory — use the original XML as-is.

Figure 5.

```
<category id="PR.AC">
  <name>Access Control</name>
  <description>Access to assets…</description>
  <subcategory id="PR.AC-1">
    <description>Identities and credentials…</description>
    <sp800-53>
      <control>AC-2</control><family>IA</family>
    </sp800-53>
  </subcategory>
  <subcategory id="PR.AC-2">…</subcategory>
  <subcategory id="PR.AC-3">…</subcategory>
  <subcategory id="PR.AC-4">
    <description>Access permissions are…</description>
    <sp800-53>
      <control>AC-2</control><control>AC-3</control>
      <control>AC-5</control><control>AC-6</control>
      <control>AC-16</control>
    </sp800-53>
  </subcategory>
  <subcategory id="PR.AC-5">…</subcategory>
</category>
```

XML representation of category PR.AC in `core.xml` showing informative references to security control AC-2. Ellipsis symbols indicate content not relevant to the example.

Figure 6 shows how `families.xml` represents security control AC-2. Baseline Tailor uses the `family` element's `name` attribute to populate the UI's "Control family" drop-down list, shown in Figure 9. After the user selects a family from the list, Baseline Tailor uses the `control` element's `number` attribute and `title` element to populate the UI's Control drop-down list, shown in Figure 10. The `default` element represents a security control's baseline impact level ("1" for low, "2" for moderate, "3" for high, and "4" if the control is not in one of the NIST SP 800-53 baselines). The `priority` element represents a security control's priority code. NIST SP 800-53 recommends that Priority 1 controls should be implemented first, followed by priority 2, and finally priority 3. Baseline tailor uses a control's `default` and `priority` sub-elements, in conjunction with the user's "Baselines" and "Priorities" checkbox selections (as shown in Figure 10), to determine whether to include the control in the "Control" drop-down list.

The control's `subcategory` elements reference all Framework Core subcategories that informatively reference the control. The `number` attributes provide these reverse references. The reverse references to PR.AC-1 and PR.AC-4 correspond to the informative references shown in Figure 5.

Figure 6.

```
<family name="ACCESS CONTROL">
  <control number="AC-1">…</control>
  <control number="AC-2">
    <title>ACCOUNT MANAGEMENT</title>
    <default>1</default>
    <priority>1</priority>
    <subcategory number="PR.AC-1"/>
    <subcategory number="PR.AC-4"/>
    <subcategory number="DE.CM-1"/>
    <subcategory number="DE.CM-3"/>
  </control>
  …
</family>
```

XML representation of "Access Control" family in `families.xml` showing cross references from security control AC-2 to Framework Core subcategories shown. Ellipsis symbols indicate content not relevant to example.

5. Baseline Tailor Usage Scenario

The flowchart in Figure 7 shows a suggested workflow for the Baseline Tailor usage scenario of using a Framework Profile and NIST SP 800-82 to support selection of NIST SP 800-53 security controls. The user begins by creating a Profile containing a set of Framework Core subcategories needed to meet a cybersecurity requirement. Next, the user considers each of the Profile's informative references. For each security control referenced, the user performs the following actions to determine how critical the security control is to achieving the Profile's outcomes:

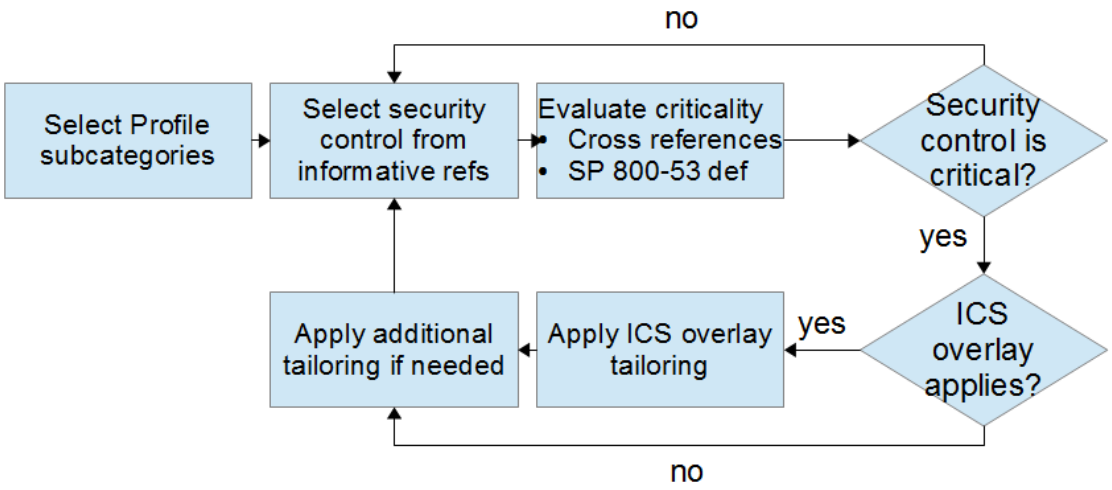
- Checks how many of the Profile's subcategories reference the security control.
- Views the security control's NIST SP 800-53 online database definition to determine relevance.

If the user deems the security control to be critical for meeting the cybersecurity requirement, the user then proceeds to tailor the security control. The user may apply the NIST SP 800-82 ICS overlay tailoring guidance, if applicable, as a starting point.

As a concrete example of the workflow in Figure 7, suppose a cybersecurity analyst wants to protect an ICS. The analyst decides to use Baseline Tailor to help determine which security controls should be selected and tailored for implementation. The analyst begins by choosing the "Protect" (PR) core function and "Access Control" (PR.AC) category in the Cyber Framework Browser tab (as shown in Figure 3). Using the Subcategory drop-down list, the analyst next looks at PR.AC's five subcategories and decides to create a Profile containing all of them. To do so, the analyst switches to the Framework Profile tab and makes the checkbox selections shown in Figure 8. Baseline Tailor creates a simple XML representation of the Profile on the fly. The Profile, a dynamic XForms model instance, is used to generate (also on the fly) XML output shown in non-editable text field at the bottom of the figure. This XML may be copy- pasted into a third-party XML authoring tool.[2]

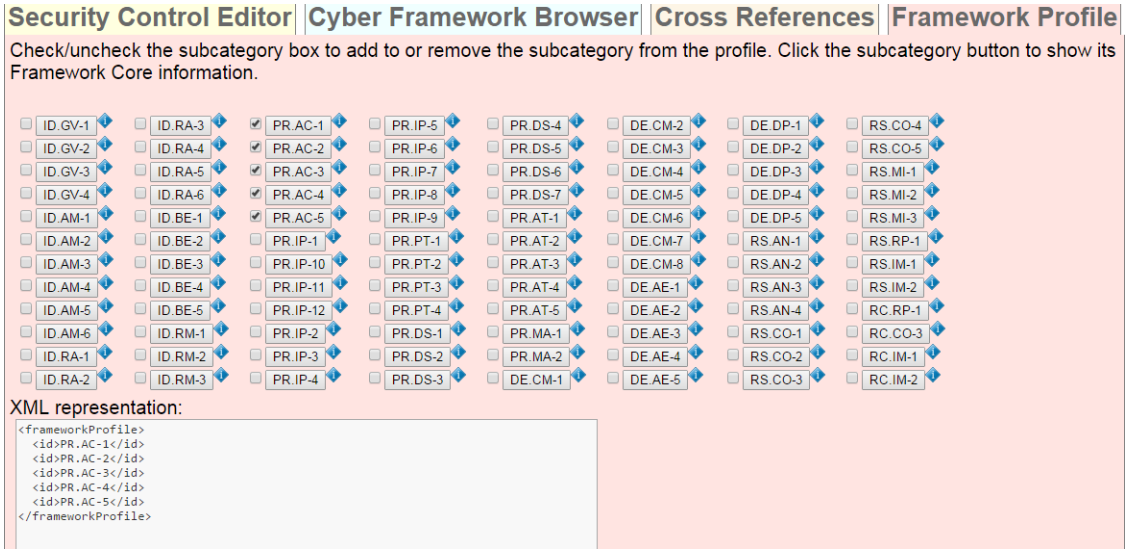
[2]Baseline Tailor's Security Control Editor tab also creates XML output on the fly. This output is generated from another dynamic model instance that encodes how the user has tailored a security control. The XML format for tailored security controls, discussed in [Lubell2015] and [Lubell2016], is both more complex and representationally richer than the simple Profile format shown in Figure 8.

Figure 7.



Workflow synthesizing Framework Core, NIST SP 800-53, and NIST SP 800-82 guidance.

Figure 8.

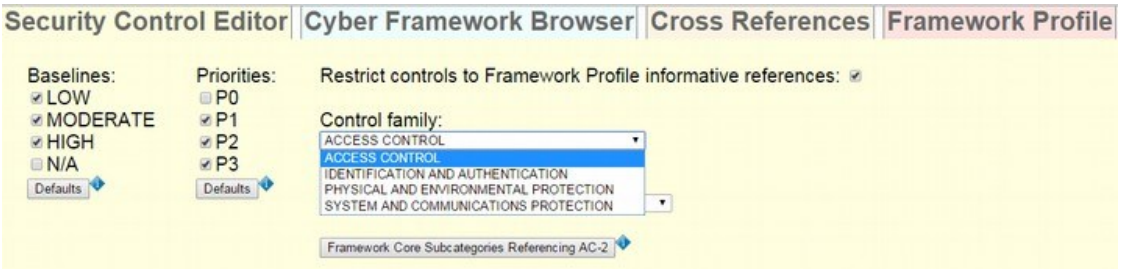


Framework Profile tab.

The analyst now switches to the Security Control Editor tab and checks a box restricting control choices to only those that are referenced by subcategories of PR.AC. As shown in Figure 9, the PR.AC subcategories reference only four of the eighteen NIST SP 800-53 control families. Now suppose the analyst selects ACCESS CONTROL from the "Control family" drop-down list, and then chooses "AC-2 – ACCOUNT MANAGEMENT" from the "Control" drop-down list populated with the subset of the Access Control family that the Profile references (Figure 10). The Security Control Editor tab now displays the UI for tailoring AC-2, the upper portion of which is shown in Figure 11.[3]

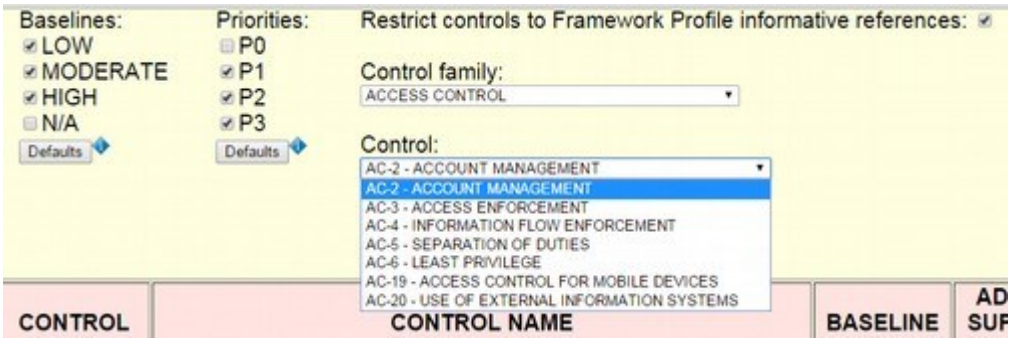
[3] [Lubell2016] discusses in detail the lower portion of the tailoring UI, which has editable text fields for adding supplemental guidance and rationale, and a non-editable text field providing XML output representing the tailored control.

Figure 9.



Control families referenced by PR.AC subcategories.

Figure 10.



Controls belonging to Access Control family that are referenced by PR.AC subcategories.

Figure 11.

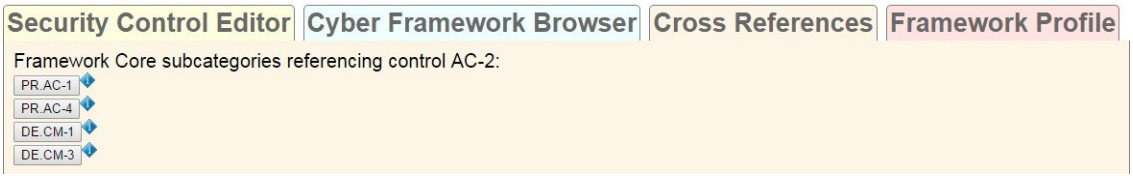
CONTROL NUMBER	CONTROL NAME *Control Enhancement Name*	BASELINE IMPACT	ADDED SUPPLE-MENTAL GUIDANCE	CONTROL BASELINES		
				LOW	MODERATE	HIGH
AC-2	**ACCOUNT MANAGEMENT**	LOW		Selected	Selected	Selected
AC-2(1)	*AUTOMATED SYSTEM ACCOUNT MANAGEMENT*	MODERATE	NO		Selected	Selected
AC-2(2)	*REMOVAL OF TEMPORARY / EMERGENCY ACCOUNTS*	MODERATE	NO		Selected	Selected
AC-2(3)	*DISABLE INACTIVE ACCOUNTS*	MODERATE	NO		Selected	Selected
AC-2(4)	*AUTOMATED AUDIT ACTIONS*	MODERATE	NO		Selected	Selected
AC-2(5)	*INACTIVITY LOGOUT*	HIGH	NO			Selected
AC-2(6)	*DYNAMIC PRIVILEGE MANAGEMENT*	N/A	NO			
AC-2(7)	*ROLE-BASED SCHEMES*	N/A	NO			
AC-2(8)	*DYNAMIC ACCOUNT CREATION*	N/A	NO			
AC-2(9)	*RESTRICTIONS ON USE OF SHARED GROUPS / ACCOUNTS*	N/A	NO			
AC-2(10)	*SHARED / GROUP ACCOUNT CREDENTIAL TERMINATION*	N/A	NO			
AC-2(11)	*USAGE CONDITIONS*	HIGH	NO			Selected
AC-2(12)	*ACCOUNT MONITORING / ATYPICAL USAGE*	HIGH	NO			Selected
AC-2(13)	*DISABLE ACCOUNTS FOR HIGH-RISK INDIVIDUALS*	HIGH	NO			Selected

Security control AC-2.

At this point, the analyst wishes to determine security control AC-2's criticality with respect to Framework Core category PR.AC. Clicking the "Framework Core Subcategories Referencing AC-2" button in Figure 9 switches to the Cross References tab, revealing that two of the five PR.AC subcategories – PR.AC-1 and PR.AC-4 – reference AC-2 (shown in Figure 12). Concluding that security control AC-2 should be selected for implementation, the analyst clicks the AC-2 button shown in the upper left of

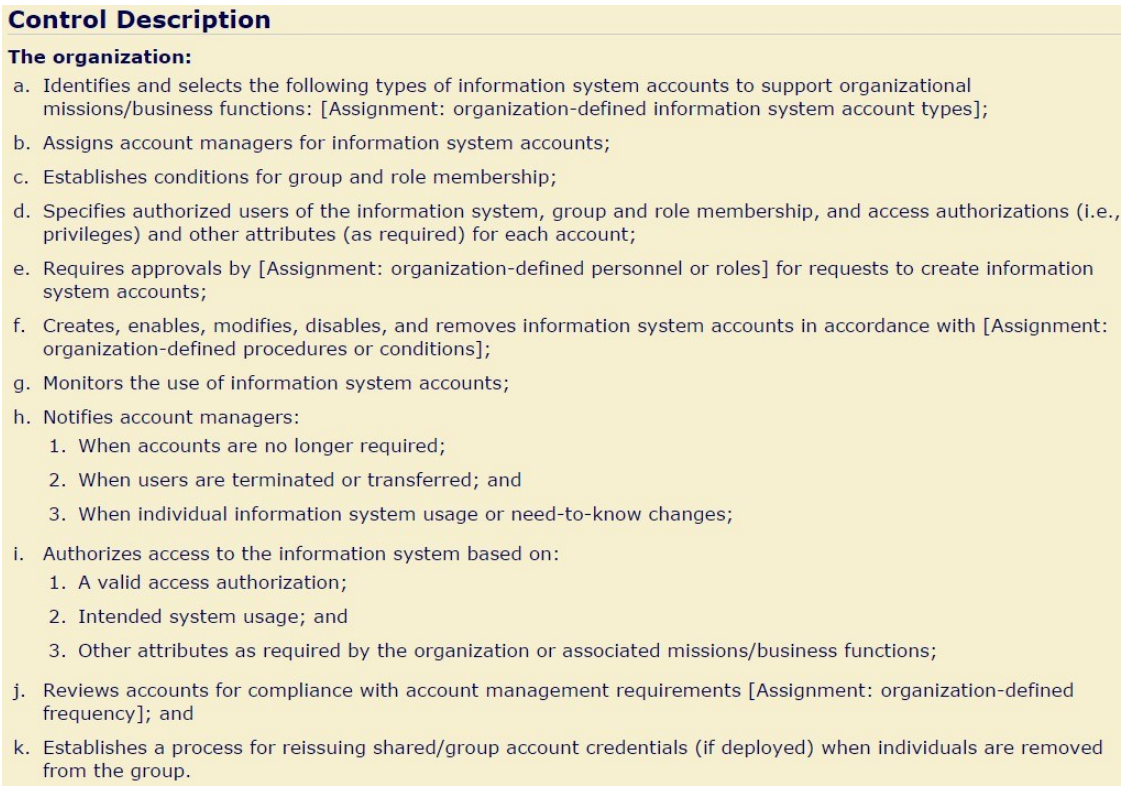
Figure 11 to look up AC-2's definition in the NIST SP 800-53 online database. Items *d*, *i*, and *k* in the AC-2 Control Description (Figure 13) are relevant to category PR.AC. The analyst therefore decides to go ahead and tailor AC-2 for the ICS.

Figure 12.



Subcategories referencing AC-2.

Figure 13.



NIST SP 800-53 online database: AC-2 description.

The analyst now clicks on the button with the factory image in Figure 11, to the right of the AC-2 button, to view AC-2's tailoring guidance in the NIST SP 800-82 ICS overlay. The overlay guidance (Figure 14) retains the same baseline allocation as NIST SP 800-53, but adds ICS-specific supplemental guidance suggesting compensating controls. Compensating controls are alternatives, for when the NIST SP 800-53 recommendations are not feasible, that provide comparable protection. The compensating controls mentioned in Figure 14 meet requirements specific to ICS. For example, an ICS may have limited network connectivity and only a small number of potential users, making physical security measures possibly more cost-effective than account management (where information processing overhead might impact performance). Using the NIST SP 800-82 guidance as a starting point, the analyst proceeds to tailor AC-2 using Baseline Tailor's Security Control Editor tab.

Figure 14.

AC-2 ACCOUNT MANAGEMENT

CNTL NO.	CONTROL NAME *Control Enhancement Name*	CONTROL BASELINES		
		LOW	MOD	HIGH
AC-2	**Account Management**	Selected	Selected	Selected
AC-2 (1)	*ACCOUNT MANAGEMENT | AUTOMATED SYSTEM ACCOUNT MANAGEMENT*		Selected	Selected
AC-2 (2)	*ACCOUNT MANAGEMENT | REMOVAL OF TEMPORARY / EMERGENCY ACCOUNTS*		Selected	Selected
AC-2 (3)	*ACCOUNT MANAGEMENT | DISABLE INACTIVE ACCOUNTS*		Selected	Selected
AC-2 (4)	*ACCOUNT MANAGEMENT | AUTOMATED AUDIT ACTIONS*		Selected	Selected
AC-2 (5)	*ACCOUNT MANAGEMENT | INACTIVITY LOGOUT / TYPICAL USAGE MONITORING*			Selected
AC-2 (11)	*ACCOUNT MANAGEMENT | USAGE CONDITIONS*			Selected
AC-2 (12)	*ACCOUNT MANAGEMENT | ACCOUNT MONITORING / ATYPICAL USAGE*			Selected
AC-2 (13)	*ACCOUNT MANAGEMENT | ACCOUNT REVIEWS*			Selected

ICS Supplemental Guidance: Example compensating controls include providing increased physical security, personnel security, intrusion detection, auditing measures.
Control Enhancement: (1, 3, 4) ICS Supplemental Guidance: Example compensating controls include employing nonautomated mechanisms or procedures.
Control Enhancement: (2) ICS Supplemental Guidance: In situations where the ICS (e.g., field devices) cannot support temporary or emergency accounts, this enhancement does not apply. Example compensating controls include employing nonautomated mechanisms or procedures.
Control Enhancement: (5) ICS Supplemental Guidance: Example compensating controls include employing nonautomated mechanisms or procedures.
Control Enhancement: (11, 12, 13) No ICS Supplemental Guidance.

NIST SP 800-82 ICS Overlay definition: AC-2.

To summarize, the scenario discussed in this section shows how a UI implemented solely with XML technologies can increase the utility of the Framework Core, NIST SP 800-53 database, and NIST SP 800-82 ICS overlay. Baseline Tailor not only provides a common UI bringing them all together, but also derives important inter-relationships. As the example showed, a Framework Profile can be used to limit the Security Control Editor tab's "Control family" and "Control" drop-down choices to the subset of NIST SP 800-53 security controls likely to be most relevant to the Profile. In addition, the Cross References tab can be used as a metric for a security control's importance with respect to the Framework Core.

6. Related Research

Previous research efforts in the areas of risk management, quality and comprehension of spreadsheet data, and the use of XPath for data integration influenced the approach described in this paper.

Linkov et al. [Linkov] studied existing risk-based guidance in the nuclear power regulation, nanotechnology, and cybersecurity fields. Defining risk as the product *threat×vulnerability×consequence*, they found that in all three cases a traditional bottom-up approach was insufficient for quantifying these three variables. Reasons why included uncertainty regarding emerging threats, lack of clear guidance for risk mitigation and determining risk tolerance, and a poor understanding of stakeholders' socio-political concerns. Linkov et al. concluded that a hybrid approach combining top-down decision making with bottom-up risk analysis can make it easier for organizations to determine and manage risk. With respect to cybersecurity, Linkov et. al observed that NIST's "Guide for Conducting Risk Assessments" [SP800-30] recommends taking an organization's risk tolerance into account when assessing risk. The Framework Profile part of the Cybersecurity Framework helps in fulfilling this recommendation by providing a means for ensuring that an organization's cybersecurity strategy, risk tolerance, and mission/business objectives are all aligned.

Numerous research efforts focused on issues with spreadsheets as a means of representing and disseminating information, a common thread being the inability of spreadsheets to capture context. Context includes information such as why content was created and how it relates to other content [OAIS]. Durusau and Hunting [Durusau], citing news reports of business calamities that were caused by errors in spreadsheet data, enumerated root causes of the errors and suggested that topic maps could help in providing the missing context information. Kohlhase et al. [Kohlhase] conducted experiments that confirmed lack of

context information as a major cause of semantic misunderstandings of data in spreadsheets. Hung et al. [Hung] developed a spreadsheet-like formula language to map spreadsheet data to a target schema and implemented the language as an Excel plug-in. Cunha et al. [Cunha2009a],[Cunha2009b], employing methods for automatically detecting functional dependencies, developed and implemented formalized approaches for improving spreadsheet quality.

Recent advances in cloud computing and web technologies have motivated researchers to investigate XPath and XPath-based languages as a means for integration of information from distributed sources. Pedersen et al. [Pedersen] used XPath as part of a formal semantic foundation for on-the-fly multidimensional data integration. The formalism uses XPath combined with a subset of the Structured Query Language (SQL) [Date]. Rennau and Grün [Rennau] determined that XQuery [XQuery] is a highly useful integration language for heterogeneous information sources, with the caveat that enhancements to XQuery and related standards are needed to improve navigational abilities for some non-XML sources.

7. Concluding Remarks

This paper presented a technical approach employing XSLT and XForms for developing a UI that integrates information from multiple sources. The original information sources may or may not be XML, and the original presentation may be either top-down or bottom-up. The Baseline Tailor software application validates the technical approach, adding value for cybersecurity professionals wishing to use the Cybersecurity Framework and NIST SP 800-53 guidance together. The `core.xml` static XForms model instance that provides the information displayed in the Cyber Framework Browser tab (Figure 3) a useful contribution in its own right since the current edition of the Cybersecurity Framework lacks a structured XML representation of the Framework Core. The Baseline Tailor software application, `core.xml`, and related XML resources are available at http://www.nist.gov/el/msid/baselinetailor.cfm.

Interestingly, Baseline Tailor was originally conceived as software only for tailoring the SP 800-53 security controls. A later version added the ability to browse the Cybersecurity Framework Core, but did not support bidirectional traversal of links between subcategories and security controls. Full integration came later, after the author began working with a team developing a Framework Profile for manufacturing systems. To incorporate guidance from the NIST SP 800-53 security control catalog and NIST SP 800-82 ICS overlay into the Manufacturing Profile, the team frequently needed to trace backwards from security controls to subcategories. This was cumbersome using the tables in the Cybersecurity Framework and NIST SP 800-53 documents. Baseline Tailor's Cross References tab made the task much easier. The team's experience before versus after the Cross References tab was added to Baseline Tailor validates the hybrid approach to risk management advocated in [Linkov].

A major limitation of the technical approach described in Section 3 is its reliance on hand-editing for semi-automated conversion of spreadsheet data to XML. It might be feasible to implement a more automated solution using the mapping language developed by Hung et al., or functional dependency detection methods from Cunha et al. A challenge with either automation approach would be getting spreadsheet authors to cooperate. A big attraction of spreadsheets as a medium for disseminating information is that authoring them is easy. Requiring authors to encode transformation logic as formulas or to think about functional dependencies makes spreadsheet production harder, although it may make life easier for spreadsheet consumers.

Acknowledgments

The author would like to thank KC Morris, Bob Lipman, Rick Candell, and the Balisage peer reviewers for their helpful comments on earlier drafts of this paper. Any remaining mistakes are the author's sole responsibility. Thanks are also due to the author's colleagues on NIST's Cybersecurity for Smart Manufacturing Systems project for their support and for many enlightening discussions.

Disclaimer

Mention of third-party or commercial products or services in this paper does not imply approval or endorsement by the National Institute of Standards and Technology, nor does it imply that such products or services are necessarily the best available for the purpose.

References

[CSF] National Institute of Standards and Technology (NIST) and United States of America. "Framework for Improving Critical Infrastructure Cybersecurity." (2014). http://www.nist.gov/cyberframework.

[CSFTool] "NIST Cybersecurity Framework (CSF) Reference Tool." http://www.nist.gov/cyberframework/csf_reference_tool.cfm. Accessed April 29, 2016.

[Cunha2009a] Cunha, Jacome, Joao Saraiva, and Joost Visser. "Discovery-Based Edit Assistance for Spreadsheets." In Symposium on Visual Languages and Human-Centric Computing (VL/HCC). 233–37. IEEE (2009).

[Cunha2009b] Cunha, Jacome, Joao Saraiva, and Joost Visser. "From Spreadsheets to Relational Databases and Back." In Proceedings of the 2009 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation, 179–88. Savannah, GA, USA (2009).

[Date] Date, Chris J., and Hugh Darwen. *A guide to the SQL Standard: a user's guide to the standard relational language SQL*. Vol. 55822. Addison-Wesley Longman (1993).

[Durusau] Durusau, Patrick, and Sam Hunting. "Spreadsheets - 90+ million End User Programmers with No Comment Tracking or Version Control." Presented at Balisage: The Markup Conference 2015, Washington, DC, August 11 - 14, 2015. In Proceedings of Balisage: The Markup Conference 2015. Balisage Series on Markup Technologies, vol. 15 (2015). 10.4242/BalisageVol15.Durusau01.

[Hung] Hung, Vu, Boualem Benatallah, and Regis Saint-Paul. "Spreadsheet-Based Complex Data Transformation." In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 1749–54 (2011).

[ISO29500] ISO/IEC 29500-1:2012. "Information technology - Document description and processing languages - Office Open XML File Formats - Part 1: Fundamentals and Markup Language Reference."

[Kohlhase] Kohlhase, Andrea, Michael Kohlhase, and Ana Guseva. "Context in Spreadsheet Comprehension." Proceedings of the Second Workshop on Software Engineering Methods in Spreadsheets. Vol. 1355. Florence, Italy: CEUR Workshop Proceedings, 21-27 (2015).

[Linkov] Linkov, Igor, Elke Anklam, Zachary A. Collier, Daniel DiMase, and Ortwin Renn. "Risk-based standards: integrating top–down and bottom–up approaches." *Environment Systems and Decisions.* 34, 134–137 (2014). 10.1007/s10669-014-9488-3.

[Lubell2014] Lubell, Joshua. "XForms User Interfaces for Small Arcane Nontrivial Datasets." Presented at Balisage: The Markup Conference 2014, Washington, DC, August 5 - 8, 2014. In *Proceedings of Balisage: The Markup Conference 2014*. Balisage Series on Markup Technologies, vol. 13 (2014). 10.4242/BalisageVol13.Lubell01.

[Lubell2015] Lubell, Joshua. "Extending the Cybersecurity Digital Thread with XForms." Presented at Balisage: The Markup Conference 2015, Washington, DC, August 11 - 14, 2015. In *Proceedings of Balisage: The Markup Conference 2015*. Balisage Series on Markup Technologies, vol. 15 (2015). 10.4242/BalisageVol15.Lubell01.

[Lubell2016] Lubell, Joshua. "Baseline Tailor User Guide." NISTIR 8130. National Institute of Standards and Technology (2016). 10.6028/NIST.IR.8130.

[NVD] "NVD - 800-53." https://web.nvd.nist.gov/view/800-53/home. Accessed April 29, 2016.

[OAIS] "Reference Model for an Open Archival Information System (OAIS)." Recommended Practice CCSDS 650.0-M-2. Consultative Committee for Space Data Systems (2012).

[Pedersen] Pedersen, Torben Bach, Dennis Pedersen, and Karsten Riis. "On-demand multidimensional data integration: toward a semantic foundation for cloud intelligence." *The Journal of Supercomputing.* 65, 217–257 (2013). 10.1007/s11227-011-0712-3.

[Rennau] Rennau, Hans-Jürgen, and Christian Grün. "XQuery as a data integration language." Presented at Balisage: The Markup Conference 2015, Washington, DC, August 11 - 14, 2015. In *Proceedings of Balisage: The Markup Conference 2015.* Balisage Series on Markup Technologies, vol. 15 (2015). 10.4242/ BalisageVol15.Rennau01.

[SP800-30] Joint Task Force Transformation Initiative. "Guide for Conducting Risk Assessments." NIST Special Publication 800-30. Revision 1 (2012). 10.6028/NIST.SP.800-30r1.

[SP800-53] Joint Task Force Transformation Initiative. "Security and Privacy Controls for Federal Information Systems and Organizations." NIST Special Publication 800-53. Revision 4 (2013). 10.6028/ NIST.SP.800-53r4.

[SP800-82] Stouffer, Keith, Victoria Pillitteri, Suzanne Lightman, Marshall Abrams, and Adam Hahn. *Guide to Industrial Control Systems (ICS) Security.* NIST Special Publication 800-82. Revision 2 (2015). 10.6028/ NIST.SP.800-82r2.

[XForms] "XForms 1.1." W3C Recommendation (2009). http://www.w3.org/TR/xforms.

[XML] "Extensible Markup Language (XML) 1.0 (Fifth Edition)." W3C Recommendation (2008). http://www.w3.org/ TR/xml.

[XPath] "XML Path Language (XPath) 3.0." W3C Recommendation (2014). http://www.w3.org/TR/xpath-30.

[XQuery] "XQuery 3.0: An XML Query Language." W3C Recommendation (2014). http://www.w3.org/TR/ xquery-30.

[XSLT] "XSL Transformations (XSLT) Version 2.0." W3C Recommendation (2007). http://www.w3.org/TR/xslt20.