STIX Version 1.2.1 Part 15: UML Model

Working Draft 01

03 August 2015

Technical Committee:

[OASIS Cyber Threat Intelligence (CTI) TC](https://www.oasis-open.org/committees/cti/)

Chair:

Richard Struse ([Richard.Struse@HQ.DHS.GOV](mailto:Richard.Struse@HQ.DHS.GOV)), [DHS Office of Cybersecurity and Communications (CS&C)](http://www.dhs.gov/office-cybersecurity-and-communications)

Editors:

Sean Barnum ([sbarnum@mitre.org](mailto:sbarnum@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Desiree Beck ([dbeck@mitre.org](mailto:dbeck@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Aharon Chernin ([achernin@soltra.com](mailto:achernin@soltra.com)), [Soltra](http://www.soltra.com/)

Rich Piazza ([rpiazza@mitre.org](mailto:rpiazza@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Additional artifacts:

This prose specification is one component of a Work Product which consists of:

* *STIX Version 1.2.1 Part 1: Overview*. [URI – added during publication]
* *STIX Version 1.2.1 Part 2: Common*. [URI]
* *STIX Version 1.2.1 Part 3: Core*. [URI]
* *STIX Version 1.2.1 Part 4: Indicator*. [URI]
* *STIX Version 1.2.1 Part 5: TTP*. [URI]
* *STIX Version 1.2.1 Part 6: Incident*. [URI]
* *STIX Version 1.2.1 Part 7: Threat Actor*. [URI]
* *STIX Version 1.2.1 Part 8: Campaign*. [URI]
* *STIX Version 1.2.1 Part 9: Course of Action*. [URI]
* *STIX Version 1.2.1 Part 10: Exploit Target*. [URI]
* *STIX Version 1.2.1 Part 11: Report*. [URI]
* *STIX Version 1.2.1 Part 12: Default Extensions*. [URI]
* *STIX Version 1.2.1 Part 13: Data Marking*. [URI]
* *STIX Version 1.2.1 Part 14: Vocabularies*. [URI]
* *STIX Version 1.2.1 Part 15: UML Model*. (this document)

Related work:

This specification is related to:

* *CybOX Version 2.1.1 (placeholder)*

Abstract:

The Structured Threat Information Expression (STIX) is a collaborative, community-driven effort to define and develop a framework for expressing cyber threat information to enable cyber threat information sharing and cyber threat analysis. The STIX framework comprises a collection of extensible component specifications along with an overarching core specification and supporting specifications. This document describes the use of UML to create a data model for STIX.

Status:

This [Working Draft](https://www.oasis-open.org/policies-guidelines/tc-process#dWorkingDraft) (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or [approved](https://www.oasis-open.org/policies-guidelines/tc-process#committeeDraft) as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document [Approval Process](https://www.oasis-open.org/policies-guidelines/tc-process#standApprovProcess) begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

URI patterns:

Initial publication URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part15-uml-model/stix-v1.2.1-csd01-part15-uml-model.docx

Permanent “Latest version” URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part15-uml-model.docx

(Managed by OASIS TC Administration; please don’t modify.)

Copyright © OASIS Open 2015. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](https://www.oasis-open.org/policies-guidelines/ipr) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

1 Introduction 4

1.1 STIX Specification Documents 4

1.2 Document Conventions 5

1.2.1 Fonts 5

1.3 Terminology 5

1.4 Normative References 5

1.5 Non-Normative References 6

2 UML Model Artifact 7

3 Data Model Conventions 8

3.1 UML Packages 8

3.2 Naming Conventions 11

3.3 UML Diagrams 11

3.3.1 Class Properties 12

3.3.2 Diagram Icons and Arrow Types 12

3.3.3 Color Coding 13

4 Conformance 14

Appendix A. Acknowledgments 15

Appendix B. Revision History 16

# Introduction

[All text is normative unless otherwise labeled]

The objective of the Structured Threat Information Expression (STIX) effort is to specify the structure and semantics of a language for capturing and characterizing cyber threat information. The normative specification of the language structure is defined in the form of a formal UML model and a set of textual specification documents that explain the UML model. The set of textual specification documents also provide clarification of language semantics that the UML model is unable to convey.

This specification document provides brief summary information on the form and use of the STIX language UML model. In addition to this textual specification document, the *STIX Version 1.2.1 Part 15: UML Model* consists of an actual digital serialization of the UML model and a set of relevant UML diagrams extracted from the UML model and used throughout the STIX language specification.

In Section **1.1** we discuss additional specification documents, in Section Error! Reference source not found.we provide document conventions, and in Section Error! Reference source not found. we provide terminology. References are given in Sections Error! Reference source not found. and Error! Reference source not found.. In Section Error! Reference source not found., we give summary information on the form of the digitally serialized UML model artifact and in Section **[enter field code here]** we provide general information and conventions for how the UML model is used to define the individual data models. Conformance information is provided in Section Error! Reference source not found.

## STIX Specification Documents

Specification documents have been written for each of the key individual data models that compose the full STIX UML model.

The [*STIX Version 1.2.1 Part 1: Overview*](#AdditionalArtifacts) document provides a comprehensive overview of the full set of STIX data models, which in addition to the nine data models mentioned in the Introduction, includes a core data model, a common data model, a cross-cutting data marking data model, various extension data models, and a set of default controlled vocabularies. [*STIX Version 1.2.1 Part 1: Overview*](#AdditionalArtifacts) also summarizes the relationship of STIX to other languages and outlines general STIX data model conventions.

**Error! Reference source not found.** illustrates the [set of specification documents](#AdditionalArtifacts) that are available. The color black is used to indicate the specification overview document, altered shading differentiates the overarching Core and Common data models from the supporting data models (default vocabularies, data marking, and extensions), and the color white indicates the component data models. This STIX Language UML Model specification document is highlighted in its associated yellow. For a list of all STIX documents and related information sources, please see [*STIX Version 1.2.1 Part 1: Overview*](#AdditionalArtifacts).

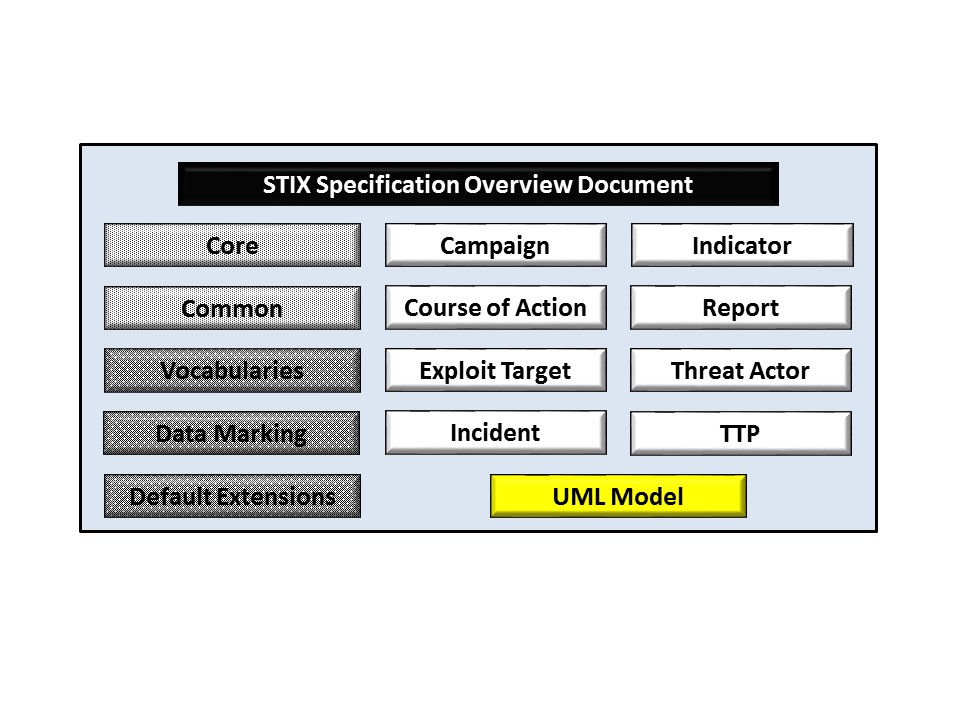


Figure 1‑1. STIX Language v1.2.1 specification documents

## Document Conventions

The following conventions are used in this document.

### Fonts

The following font and font style conventions are used in the document:

* Capitalization is used for STIX high level concepts.

Examples: Indicator, Course of Action, Threat Actor

* The Courier New font is used for writing UML objects.

Examples: RelatedIndicatorsType, stixCommon:StatementType

Note that all high level concepts have a corresponding UML object. For example, the Course of Action high level concept is associated with a UML class named, CourseOfActionType.

* The ‘*italic’* font (withsingle quotes) is used for noting actual, explicit values for STIX Language properties. The *italic* font (without quotes) is used for noting example values.

Example:  *‘PackageIntentVocab-1.0,’ high, medium, low*

## Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

## Normative References

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.

## Non-Normative References

**[****GitHub-IO]** STIX – Structured Threat Information Expression | STIX Project Documentation. (n.d.). The MITRE Corporation. [Online]. Available: <http://stixproject.github.io/>. Accessed Aug. 23, 2015.

**[****STIX-W**] Barnum, S., “Standardizing Cyber Threat Intelligence with the Structured Threat Information eXpression (STIXTM),” The MITRE Corporation, Bedford MA, Feb. 20, 2014. [Online]. Available: <http://stixproject.github.io/getting-started/whitepaper/>.

**[****UML-2.4.1**] Documents associated with Unified Modeling Language (UML), V2.4.1. (Aug. 2011). The Object Management Group (OMG). [Online]. Available: <http://www.omg.org/spec/UML/2.4.1/>.

**[XMI**] Documents associated with XMI Version 2.1. (September 2005). The Object Management Group (OMG). [Online]. Available: <http://www.omg.org/spec/XMI/2.1/>

.

# UML Model Artifact

The STIX UML model is formally represented in the form of a digital serialization using the XML Metadata Interchange (XMI) language, version x.x. [[XMI]](#XMI). The XMI language is intended to be an open standardized form supporting the expression of UML models in a non-proprietary manner. In reality, many UML modeling tools tend to include some proprietary elements in their XMI output. The STIX UML model was produced using Rational Software Architect (RSA) version 9.1, a product of the IBM Corporation. Effort has been made to minimize the level of proprietary content (from the RSA tool) in the XMI serialization but it should be noted that some portion may still remain.

In addition to the XMI serialization of the UML model, a set of relevant UML diagrams extracted from the UML model and leveraged throughout the STIX language specification documents are also provided. These diagrams are provided both in primitive (graphical modeling framework [GMF]) and rastered (portable network graphics [PNG]) forms.

# Data Model Conventions

The following general information and conventions are used to define the individual data models in UML.

## UML Packages

Each STIX data model is captured in a different UML package (e.g., Core package, Campaign package, etc.). To refer to a particular class of a specific package, we use the format package\_prefix:class, where package\_prefix corresponds to the appropriate UML package. **Table 3‑1** lists the packages used throughout the STIX data model specification documents, along with the prefix notation and an example.

Table ‑. Package prefixes used by the STIX Language

|  |  |
| --- | --- |
| Package | STIX Core |
| **Prefix** | **stix** |
| Description | The STIX Core data model defines a STIX Package that encompasses all other objects of STIX. |
| Example | stix:TTPsType |
|  | |
| Package | STIX Common |
| **Prefix** | **stixCommon** |
| Description | The STIX Common data model defines classes that are shared across the various STIX data models. |
| Example | stixCommon:ConfidenceType |
|  |  |
| Package | STIX Data Marking |
| **Prefix** | **marking** |
| Description | The STIX Data Marking data model enables data markings to be used. |
| Example | marking:MarkingType |
|  | |
| Package | STIX Default Vocabularies |
| **Prefix** | **stixVocabs** |
| Description | The STIX default vocabularies define the classes for default controlled vocabularies used within STIX. |
| Example | stixVocabs:MalwareTypeVocab |
|  | |
| Package | Packages used in STIX Default Extensions |
| **Prefix** | **a (ciq address); capec; ciq; stix-ciqidentity; maec; tlpMarking; cvrf; ioc; oval-def; oval-var** |
| Description | Various packages are used by STIX extensions. Details are given in [*STIX Version 1.2.1 Part 12: Default Extensions*](#Additional_Artifacts). |
| Example | capec:Attack\_PatternType |
|  |  |
| Package | STIX Basic Data Types |
| **Prefix** | **basicDataTypes** |
| Description | The STIX Basic Data Types data model defines the types used within STIX. |
| Example | basicDataTypes:URI |
|  |  |
| Package | STIX Indicator |
| **Prefix** | **indicator** |
| Description | The STIX Indicator data model conveys specific Observable patterns combined with contextual information intended to represent artifacts and/or behaviors of interest within a cyber security context. |
| Example | indicator:TestMechanismType |
|  |  |
| Package | STIX Incident |
| **Prefix** | **incident** |
| Description | The STIX Incident data model captures discrete instances of a specific set of observed events or properties affecting an organization. |
| Example | incident:AffectedAssetType |
|  |  |
| Package | STIX TTP |
| **Prefix** | **ttp** |
| Description | The STIX TTP data model captures the behavior or modus operandi of cyber adversaries. |
| Example | ttp:AttackPatternType |
|  |  |
| Package | STIX Campaign |
| **Prefix** | **campaign** |
| Description | The STIX Campaign data model encompasses one or more Threat Actors pursuing an Intended Effect as observed through sets of Incidents and/or TTP, potentially across organizations. |
| Example | campaign:AttributionType |
|  |  |
| Package | STIX Threat Actor |
| **Prefix** | **ta** |
| Description | The STIX Threat Actor data model captures characterizations of malicious actors (or adversaries) representing a cyber attack threat including presumed intent and historically observed behavior. |
| Example | ta:ObservedTTPsType |
|  |  |
| Package | STIX Exploit Target |
| **Prefix** | **et** |
| Description | The STIX Exploit Target data model conveys a vulnerability or weakness in software, systems, networks or configurations that is targeted for exploitation by the TTP of a Threat Actor. |
| Example | et:ConfigurationType |
|  |  |
| Package | STIX Course of Action |
| **Prefix** | **coa** |
| Description | The STIX Course of Action data model conveys specific measures to be taken to address threats whether they are corrective or preventative to address Exploit Targets, or responsive to counter or mitigate the potential impacts of Incidents. |
| Example | coa:StructuredCOAType |
|  | |
| Package | STIX Report |
| **Prefix** | **report** |
| Description | The STIX Report defines a contextual wrapper for a grouping of STIX content, which could include content specified using any of the other eight top-level constructs, or even other related Reports. |
| Example | report:RelatedReportsType |
|  |  |
|  | CybOX Core |
| **Prefix** | **cybox** |
| Description | The [CybOX](#RelatedWork) core data model defines the core constructs used in CybOX. |
| Example | cybox:ObservablesType |

## Naming Conventions

The UML classes, enumerations, and properties defined in STIX follow the particular naming conventions outlined in Table 3‑2[[1]](#endnote-1).

Table ‑. Naming formats of different object types

|  |  |  |
| --- | --- | --- |
| **Object Type** | **Format** | **Example** |
| Class | CamelCase ending with “Type” | IndicatorBaseType |
| Property (simple) | Lowercase with underscores between words | capec\_id |
| Property (complex) | Capitalized with underscores between words | Associated\_Actor |
| Enumeration | CamelCase ending with “Enum” or “Type | DateTimePrecisionEnum; IndicatorVersionType |
| Enumeration value | *varies* | Flash drive; Public Disclosure; Externally-Located |
| Data type | CamelCase or if the words are acroynms, all capitalized with underscores between words. | PositiveInteger; CVE\_ID |

## UML Diagrams

This document indicates how UML diagrams are used to visually depict relationships between STIX Language constructs in the rest of the specification. Note that the example diagrams have been extracted directly from the full UML model for STIX; they have not been constructed purely for inclusion in this or the other specification documents.  Typically, diagrams are included where the visualization of its relationships between classes is useful for illustration purposes.  This implies that there will be very few diagrams for classes whose only properties are either a data type or a class from the STIX Common data model.  All data models include a top-level diagram (see Figure 3‑1).

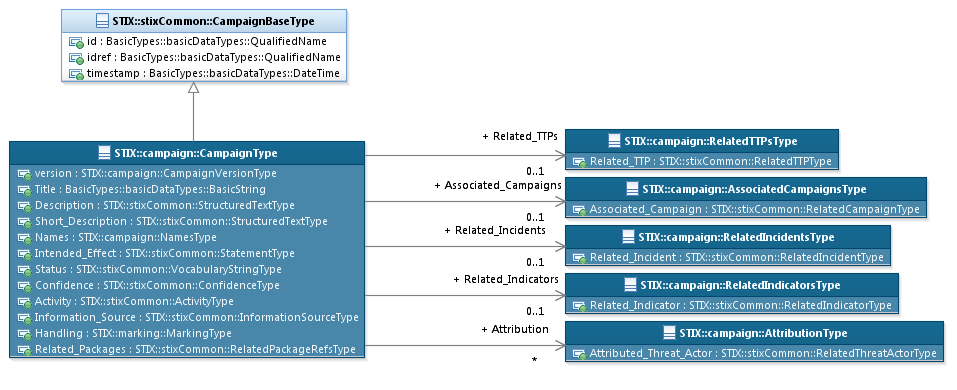


Figure ‑. Top-level Package Diagram

In UML diagrams, classes are often presented with their attributes elided, to avoid clutter. The fully described class can usually be found in a related diagram. A class presented with an empty section at the bottom of the icon indicates that there are no attributes other than those that are visualized using associations (see Figure 3‑2).

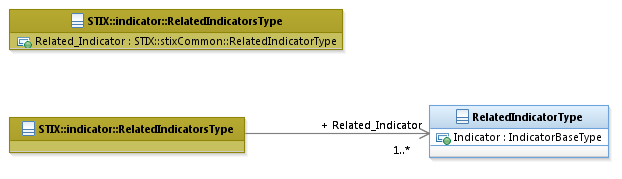


Figure ‑. Different presentations of class attributes

### Class Properties

Generally, a class property can be shown in a UML diagram as either an attribute or an association (i.e., the distinction between attributes and associations is somewhat subjective). In order to make the size of UML diagrams in the specifications manageable, we have chosen to capture most properties as attributes and to capture only higher level properties as associations, especially in the main top-level component diagrams. In particular, we will always capture properties of UML data types as attributes. For example, properties of a class that are identifiers, titles, and timestamps will be represented as attributes.

### Diagram Icons and Arrow Types

Diagram icons are used in a UML diagram to indicate whether a shape is a class, enumeration or data type, and decorative icons are used to indicate whether an element is an attribute of a class or an enumeration literal. In addition, two different arrow styles indicate either a directed association relationship (regular arrowhead) or a generalization relationship (triangle-shaped arrowhead). The icons and arrow styles we use are shown and described in **Table 3‑3**.

Table ‑. UML diagram icons

|  |  |
| --- | --- |
| **Icon** | **Description** |
| cid:image003.gif@01D05428.2B30AE20 | This diagram icon indicates a class. If the name is in italics, it is an abstract class. |
|  | This diagram icon indicates an enumeration. |
|  | This diagram icon indicates a data type. |
|  | This decorator icon indicates an attribute of a class. The green circle means its visibility is public. If the circle is red or yellow, it means its visibility is private or protected. |
|  | This decorator icon indicates an enumeration literal. |
|  | This arrow type indicates a directed association relationship. |
|  | This arrow type indicates a generalization relationship. |

### Color Coding

The shapes of the UML diagrams are color coded to indicate the data model associated with a class. The colors used in the collection of specification documents via exemplars are illustrated in **Figure 3‑1**.



Figure ‑. Data model color coding

# Conformance

Implementations have discretion over which parts (components, properties, extensions, controlled vocabularies, etc.) of STIX they implement (e.g., Indicator/Suggested\_COAs).

[1] Conformant implementations must conform to all normative structural specifications of the UML model that apply to the portions of STIX they implement (e.g., Implementers of the entire TTP component must conform to all normative structural specifications of the UML model regarding the TTP component).

[2] Conformant implementations are free to ignore normative structural specifications of the UML model that do not apply to the portions of STIX they implement (e.g., Non-implementers of any particular properties of the TTP component are free to ignore all normative structural specifications of the UML model regarding those properties of the TTP component).

The conformance section of this document is intentionally broad and attempts to reiterate what already exists in this document. The STIX 1.2 Specifications, which this specification is based on, did not have a conformance section. Instead, the STIX 1.2 Specifications relied on normative statements and the non-mandatory implementation of STIX profiles. STIX 1.2.1 represents a minimal change from STIX 1.2, and in that spirit no requirements have been added, modified, or removed by this section.

1. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Dean Thompson, Australia and New Zealand Banking Group (ANZ Bank)

Bret Jordan, Blue Coat Systems, Inc.

Adnan Baykal, Center for Internet Security (CIS)

Jyoti Verma, Cisco Systems

Liron Schiff, Comilion (mobile) Ltd.

Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)

Richard Struse, DHS Office of Cybersecurity and Communications (CS&C)

Marlon Taylor, DHS Office of Cybersecurity and Communications (CS&C)

David Eilken, Financial Services Information Sharing and Analysis Center (FS-ISAC)

Sarah Brown, Fox-IT

Ryusuke Masuoka, Fujitsu Limited

Eric Burger, Georgetown University

Jason Keirstead, IBM

Paul Martini, iboss, Inc.

Jerome Athias, Individual

Terry MacDonald, Individual

Alex Pinto, Individual

Patrick Maroney, Integrated Networking Technologies, Inc.

Wouter Bolsterlee, Intelworks BV

Joep Gommers, Intelworks BV

Sergey Polzunov, Intelworks BV

Rutger Prins, Intelworks BV

Andrei Sîrghi, Intelworks BV

Raymon van der Velde, Intelworks BV

Jonathan Baker, MITRE Corporation

Sean Barnum, MITRE Corporation

Mark Davidson, MITRE Corporation

Ivan Kirillov, MITRE Corporation

Jon Salwen, MITRE Corporation

John Wunder, MITRE Corporation

Mike Boyle, National Security Agency

Jessica Fitzgerald-McKay, National Security Agency

Takahiro Kakumaru, NEC Corporation

John-Mark Gurney, New Context Services, Inc.

Christian Hunt, New Context Services, Inc.

Daniel Riedel, New Context Services, Inc.

Andrew Storms, New Context Services, Inc.

John Tolbert, Queralt, Inc.

Igor Baikalov, Securonix

Bernd Grobauer, Siemens AG

Jonathan Bush, Soltra

Aharon Chernin, Soltra

Trey Darley, Soltra

Paul Dion, Soltra

Ali Khan, Soltra

Natalie Suarez, Soltra

Cedric LeRoux, Splunk Inc.

Brian Luger, Splunk Inc.

Crystal Hayes, The Boeing Company

Brad Butts, U.S. Bank

Mona Magathan, U.S. Bank

Adam Cooper, United Kingdom Cabinet Office

Mike McLellan, United Kingdom Cabinet Office

Chris O'Brien, United Kingdom Cabinet Office

Julian White, United Kingdom Cabinet Office

Anthony Rutkowski, Yaana Technologies, LLC

The authors would like to thank the STIX Community for its input and help in reviewing this document.

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Editors** | **Changes Made** |
| wd01 | 11 September 2015 | Sean Barnum Desiree Beck Aharon Chernin Rich Piazza | Initial authored draft |

1. These choices were made for the XML schema to differentiate XML attributes and elements. Although such distinctions are not made in the UML model, we kept the naming convention for consistency. We expect that eventually, the names of the UML model will be made uniform. [↑](#endnote-ref-1)