STIX™ 1.1.1
COURSE OF ACTION SPECIFICATION (V1.1.1)

MAY 11, 2015

The Structured Threat Information eXpression (STIX[TM]) framework defines eight core constructs and the relationships between them for the purposes of modeling cyber threat information and enabling cyber threat information analysis and sharing. This specification document defines the Course of Action construct, which 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.

Acknowledgements

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

Trademark Information

STIX, the STIX logo, and CybOX are trademarks of The MITRE Corporation. All other trademarks are the property of their respective owners.

Warnings

MITRE PROVIDES STIX "AS IS" AND MAKES NO WARRANTY, EXPRESS OR IMPLIED, AS TO THE ACCURACY, CAPABILITY, EFFICIENCY, MERCHANTABILITY, OR FUNCTIONING OF STIX. IN NO EVENT WILL MITRE BE LIABLE FOR ANY GENERAL, CONSEQUENTIAL, INDIRECT, INCIDENTAL, EXEMPLARY, OR SPECIAL DAMAGES, RELATED TO STIX OR ANY DERIVATIVE THEREOF, WHETHER SUCH CLAIM IS BASED ON WARRANTY, CONTRACT, OR TORT, EVEN IF MITRE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.[1]

Feedback

The STIX development team welcomes any feedback regarding this document. Please send comments, questions, or suggestions to stix@mitre.org.[2]

---

[1] For detailed information see [TOU].
[2] For more information about the STIX Language, please visit [STIX].

Table of Contents

1 Introduction

The Structured Threat Information eXpression (STIX™) framework defines eight top-level component data models: Observable[3], Indicator, Incident, TTP, ExploitTarget, CourseOfAction, Campaign, and ThreatActor. This document serves as the specification for the STIX Course of Action Version 1.1.1 data model.

As defined within the STIX language, a Course of Action (COA) characterizes a specific measure that could be taken in regards to a threat. These measures may be corrective or preventative to address Exploit Targets, or responsive to counter or mitigate the potential impacts of Incidents. They are typically cyber in nature but are not explicitly constrained to be so. More specifically, a Course of Action is fundamentally a characterization of the action through a title, description, type and structured observable parameters as well as contextual information such as objective, likely impact, likely cost, estimated efficacy and its relevant stage in cyber threat management (e.g., remedy of an ExploitTarget or response to an Incident).

In Section 1.1 we discuss STIX specification documents, and in Section 1.2 we give document conventions. In Section 2, we give background information necessary to fully understand the Course of Action data model, and we present the Course of Action data model specification details in Section 3. References are provided in the final section.

1.1   STIX Specification Documents

The STIX specification consists of a formal UML model and a set of textual specification documents that explain the UML model. Specification documents have been written for each of the key individual data models that compose the full STIX UML model.

The STIX specification overview document provides a comprehensive overview of the full set of STIX data models [STIX$_O$], which in addition to the eight top-level component 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$_O$] also summarizes the relationship of STIX to other languages, and outlines general STIX data model conventions.

Figure 1-1 illustrates the set of specification documents 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. The Observable component data model is shown as an oval shape to indicate that it is defined as a CybOX specification (see [STIX$_O$] for details). This Course of Action

---

[3] The CybOX Observable data model is actually defined in the CybOX Language, not in STIX.

specification document is highlighted in its associated color (see Section 1.2.4.3). For a list of all STIX documents and related information sources, please see [STIX$_O$].
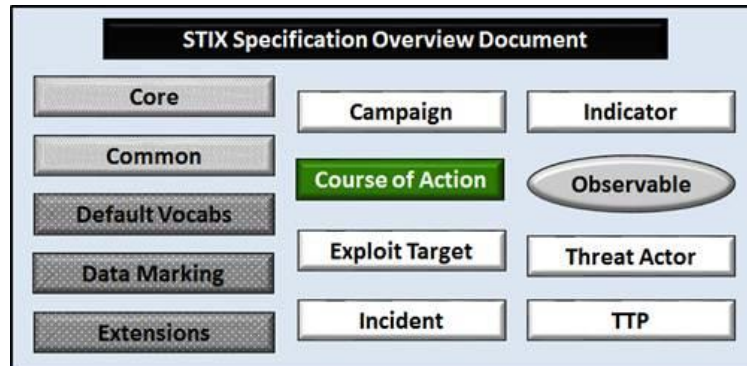
Figure 1-1. STIX Language v1.1.1 specification documents

All specification documents can be found on this STIX Website [STIX-SPECS].

1.2   Document Conventions

The following conventions are used in this document.

1.2.1   Keywords

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 *RFC 2119* [RFC2119].

1.2.2   Fonts

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

- Capitalization is used for STIX high level concepts, which are defined in the STIX Specification Overview [STIX$_O$].

  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 (with single 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.*

1.2.3   UML Package References

Each STIX data model is captured in a different UML package (e.g., Core package, Campaign package, etc.) where the packages together compose the full STIX UML model. 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. The STIX[TM] 1.1.1 Specification Overview document [STIX$_O$] contains a list of the packages used by the Course of Action data model, along with the associated prefix notations, descriptions, examples.

Note that in this specification document, we do not explicitly specify the package prefix for any classes that originate from the Course of Action data model.

1.2.4   UML Diagrams

This specification makes use of UML diagrams to visually depict relationships between STIX Language constructs. Note that the diagrams have been extracted directly from the full UML model for STIX; they have not been constructed purely for inclusion in the specification documents. Typically, diagrams are included for the primary class of a data model, and for any other class where the visualization of its relationships between other classes would be useful. 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. Other diagrams that are included correspond to classes that specialize a superclass and abstract or generalized classes that are extended by one or more subclasses.

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.

1.2.4.1 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

3

example, properties of a class that are identifiers, titles, and timestamps will be represented as attributes.

1.2.4.2 Diagram Icons and Arrow Types

Diagram icons are used in a UML diagram to indicate whether a shape is a class, enumeration, or a 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 1-1.

Table 1-1. UML diagram icons

Icon	Description
	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.

1.2.4.3 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 Course of Action specification are illustrated in Figure 1-2.
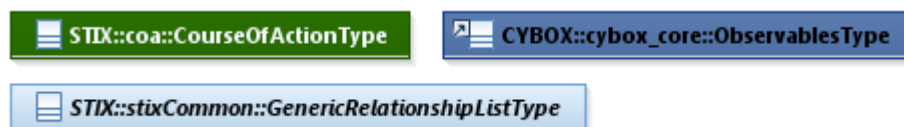
Figure 1-2.   Data model color coding

4

1.2.5   Property Table Notation

Throughout Section 2.2, tables are used to describe the properties of each data model class. Each property table consists of a column of names to identify the property, a type column to reflect the datatype of the property, a multiplicity column to reflect the allowed number of occurrences of the property, and a description column that describes the property. Package prefixes are provided for classes outside of the Course of Action data model (see Section 1.2.3).

Note that if a class is a specialization of a superclass, only the properties that constitute the specialization are shown in the property table (i.e., properties of the superclass will not be shown). However, details of the superclass may be shown in the UML diagram.

In addition, properties that are part of a "choice" relationship (e.g., Prop1 OR Prop2 is used but not both) will be denoted by a unique letter subscript (e.g., API_Call$_A$, Code$_B$) and single logic expression in the Multiplicity column. For example, if there is a choice of property `API_Call`$_A$ and `Code`$_B$, the expression "A(1)|B(0..1)" will indicate that the `API_Call` property can be chosen with multiplicity 1 or the `Code` property can be chosen with multiplicity 0 or 1.

1.2.6   Property and Class Descriptions

Each class and property defined in STIX is described using the format, "The X property <u>verb</u> Y." For example, in the specification for the STIX Campaign, we write, "The `id` property <u>specifies</u> a globally unique identifier for the Campaign instance." In fact, the verb "specifies" could have been replaced by any number of alternatives: "defines," "describes," "contains," "references," etc.

However, we thought that using a wide variety of verb phrases might confuse a reader of a specification document because the meaning of each verb could be interpreted slightly differently. On the other hand, we didn't want to use a single, generic verb, such as "describes," because although the different verb choices may or may not be meaningful from an implementation standpoint, a distinction could be useful to those interested in the modeling aspect of STIX.

Consequently, we have chosen to use the three verbs, defined as follows, in class and property descriptions:

| Verb	STIX Definition
<u>captures</u>	Used to record and preserve information without implying anything about the structure of a class or property. Often used for properties that encompass general content. This is the least precise of the three verbs.
	Examples: The `Source` property characterizes the source of the sighting information. Examples of details <u>captured</u> include identitifying

5

	characteristics, time-related attributes, and a list of the tools used to collect the information. The `Description` property <u>captures</u> a textual description of the Indicator.
<u>characterizes</u>	Describes the distinctive nature or features of a class or property. Often used to describe classes and properties that themselves comprise one or more other properties.
	Examples: The `Confidence` property <u>characterizes</u> the level of confidence in the accuracy of the overall content captured in the Incident. The `ActivityType` class <u>characterizes</u> basic information about an activity a defender might use in response to a Campaign.
<u>specifies</u>	Used to clearly and precisely identify particular instances or values associated with a property. Often used for properties that are defined by a controlled vocabulary or enumeration; typically used for properties that take on only a single value.
	Example: The `version` property <u>specifies</u> the version identifier of the STIX Campaign data model used to capture the information associated with the Campaign.

6

2 Background Information

In this section, we provide high level information about the Course of Action data model that is necessary to fully understand the Course of Action data model specification details given in Section 3.

2.1 Course of Action-Related Component Data Models

As will be explicitly detailed in Section 3, a STIX Course of Action leverages the Observables data model (as indicated by the outward-oriented arrow) which is defined with the CybOX Language. Figure 2-1 illustrates the relationship between the Course of Action and the other core constructs. As stated in Section 1.1, each of these components is defined in a separate specification document.
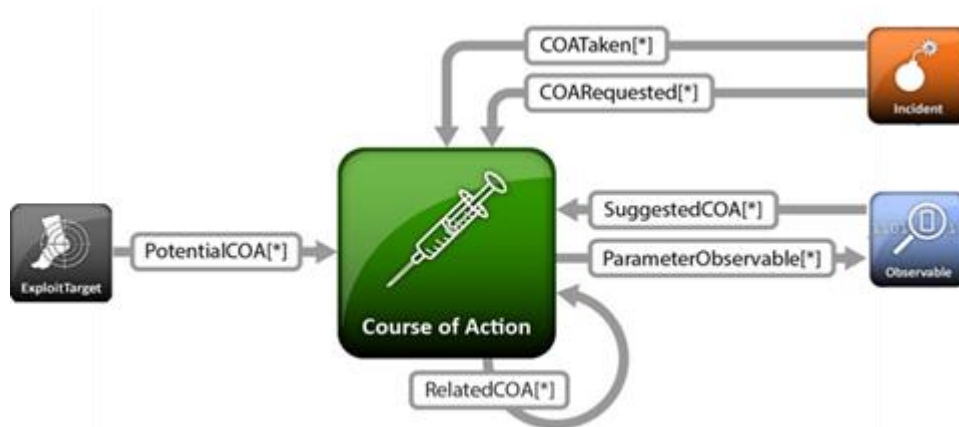
Figure 2-1. High level view of the Course of Action data model

In this section, we give a high level summary of the relationship between the Course of Action data model and the other components to which a Course of Action may refer. We also make note of the fact that the Course of Action data model can be self-referential. Other relationships are defined in the specification of the component that they originate from.

- **Course of Action**

  The Course of Action data model is self-referential, enabling one Course of Action to reference other Course of Actions that are asserted to be related. Self-referential relationships between Courses of Action may indicate general associativity or can be used to indicate relationships beween different versions of the same Course of Action.

- **Observable**

  A STIX Observable (as defined with the CybOX Language[4]) represents stateful properties or measurable events pertinent to the operation of computers and networks. Implicit in this is a practical need for descriptive capability of two forms of observables: "observable instances" and "observable patterns." Observable instances represent actual specific observations that took place in the cyber domain. The property details of this observation are specific and unambiguous. Observable patterns represent conditions for a potential observation that may occur in the future or may have already occurred and exists in a body of observable instances. These conditions may be anything from very specific concrete patterns that would match very specific observable instances to more abstract generalized patterns that have the potential to match against a broad range of potential observable instances.

  The Course of Action data model leverages the Observable data model to specify observable patterns to be used as structured parameters for the action specified in the Type property (e.g. a structured characterization of an outbound network connection to a particular IP address that when combined with a Type="Block" unambiguously describes an action of blocking such traffic).

---

[4] CybOX specification documents will be created after STIX specification documents are completed.

3 STIX Course of Action Data Model

The primary class of the STIX Course of Action package is the `CourseOfActionType` class, which characterizes a cyber threat-relevant course of action through informative (title and description), formally structured (type and parameter observables) and contextual (objective, efficacy, impact, cost) properties. Similar to the primary classes of all of the component data models in STIX, the `CourseOfActionType` class extends a base class defined in the STIX Common data model; more specifically, it extends the `CourseOfActionBaseType` base class, which provides the essential identifier (`id`) and identifier reference (`idref`) properties.

The relationship between the `CourseOfActionType` class and the `CourseOfActionBaseType` base class, as well as the properties of the `CourseOfActionType` class, are illustrated in the UML diagram given in Figure 3-1.
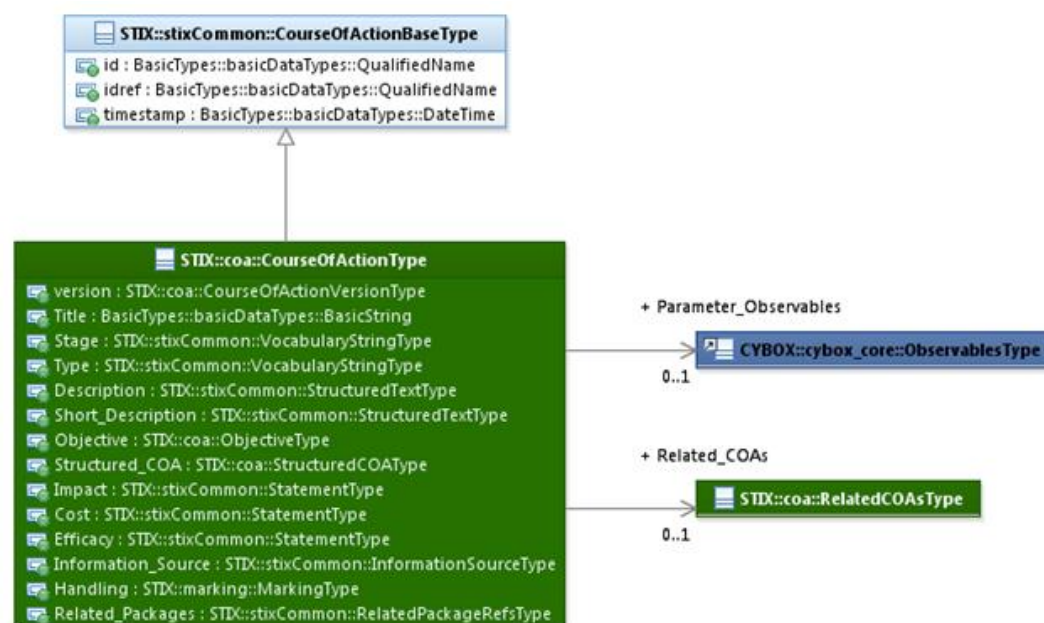
Figure 3-1. UML diagram of the `CourseOfActionType` class

9

The property table, which includes property descriptions and corresponds to the UML diagram given in Figure 3-1, is provided in Table 3-1.

All classes defined in the Course of Action data model are described in detail in Sections 3.1 through Section 3.4. Details are not provided for classes defined in non-Course of Action data models; instead, the reader is referred to the corresponding data model specification as indicated by the package prefix specified in the Type column of the table.

Table 3-1. Properties of the `CourseOfActionType` class

Name	Type	Multiplicity	Description
version	`CourseOfActionVersionType`	0..1	The `version` property specifies the version number of the STIX Course of Action data model used to capture the information associated with the Course of Action.
Title	`basicDataTypes:BasicString`	0..1	The `Title` property captures a title for the Course of Action and reflects what the content producer thinks the Course of Action as a whole should be called. The `Title` property is typically used by humans to reference a particular Course of Action; however, it is not suggested for correlation.
Stage	`stixCommon:` `VocabularyStringType`	0..1	The `Stage` property specifies what stage in the cyber threat management lifecycle this Course of Action is relevant to. Examples of potential stages include *remedy* and *response* (these specific values are only provided to help explain the property: they are neither recommended values nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary value or may constrain the set of possible values by referencing an externally-defined vocabulary or leveraging a formally defined vocabulary extending from the `stixCommon:ControlledVocabularyStringType` class. The STIX default vocabulary class for use in the property is '*COAStageVocab-1.0*'.

Type	stixCommon: VocabularyStringType	0..1	The `Type` property specifies the type of action to be taken. Examples of potential types include *redirection*, *eradication* and *public disclosure* (these specific values are only provided to help explain the property: they are neither recommended values nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary value or may constrain the set of possible values by referencing an externally-defined vocabulary or leveraging a formally defined vocabulary extending from the `stixCommon:ControlledVocabularyStringType` class. The STIX default vocabulary class for use in the property is *'CourseOfActionTypeVocab-1.0'*.
Description	stixCommon: StructuredTextType	0..1	The `Description` property captures a textual description of the Course of Action. Any length is permitted. Optional formatting is supported via the `structuring_format` property of the `StructuredTextType` class.
Short_Description	stixCommon: StructuredTextType	0..1	The `Short_Description` property captures a short textual description of the Course of Action. This property is secondary and should only be used if the `Description` property is already populated and another, shorter description is available.
Objective	ObjectiveType	0..1	The `Objective` property characterizes the results that this Course of Action is intended to achieve.
Parameter_Observables	cybox:ObservablesType	0..1	The `Parameter_Observables` property enables the specification of structured technical parameters to this Course of Action expressed using the CybOX Language. It is intended that the combination of the Course of Action Type and the Parameter_Observables could be used to define unambiguous and potentially automated courses of action.
Structured_COA	StructuredCOAType	0..1	The `Structured_COA` property characterizes an alternative actionable structured representation for the Course of Action potentially for automated consumption and implementation. Its underlying abstract class MUST be extended to enable the expression of a structured Course of Action.

Impact	`stixCommon:StatementType`	0..1	The `Impact` property characterizes the estimated impact of applying a Course of Action to achieve its targeted objective, which includes a `Value` property that specifies the level of impact. Examples of potential levels include *high*, *medium*, and *low* (these specific values are only provided to help explain the `Value` property: they are neither recommended values nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary value or may constrain the set of possible levels by referencing an externally-defined vocabulary. The STIX default vocabulary class for use in the `Value` property is '*HighMediumLowVocab-1.0.*'
Cost	`stixCommon:StatementType`	0..1	The `Cost` property characterizes the estimated cost for applying a Course of Action to achieve its targeted objective, which includes a `Value` property that specifies the level of cost. Examples of potential levels include *high*, *medium*, and *low* (these specific values are only provided to help explain the `Value` property: they are neither recommended values nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary value or may constrain the set of possible levels by referencing an externally-defined vocabulary. The STIX default vocabulary class for use in the `Value` property is '*HighMediumLowVocab-1.0.*'
Efficacy	`stixCommon:StatementType`	0..1	The `Efficacy` property characterizes a measure of the likely effectiveness of a Course of Action to achieve its targeted objective, which includes a `Value` property that specifies the level of effectiveness. Examples of potential levels include *high*, *medium*, and *low* (these specific values are only provided to help explain the `Value` property: they are neither recommended values nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary value or may constrain the set of possible levels by referencing an externally-defined vocabulary. The STIX default vocabulary class for use in the `Value` property is '*HighMediumLowVocab-1.0.*'

Information_Source	stixCommon: InformationSourceType	0..1	The `Information_Source` property characterizes the source of the Course of Action information. Examples of details captured include identitifying characteristics, time-related attributes, and a list of tools used to collect the information.
Handling	marking:MarkingType	0..1	The `Handling` property specifies the appropriate data handling markings for the properties of this Course of Action. The marking scope is limited to the Course of Action and the content it contains. Note that data handling markings can also be specified at a higher level.
Related_COAs	RelatedCOAsType	0..1	The `Related_COAs` property specifies a set of one or more other Course of Actions related to this Course of Action.
Related_Packages	stixCommon: RelatedPackagesRefsType	0..1	The `Related_Packages` property specifies a set of one or more STIX Packages that are related to the Course of Action.

3.1   CourseOfActionVersionType Enumeration

The `CourseOfActionVersionType` enumeration is an inventory of all possible versions of the Course of Action data model, all of which are valid in STIX Version 1.1.1. The enumeration literals are given in Table 3-2.

Table 3-2. Literals of the `CourseOfActionVersionType` enumeration

Enumeration Literal	Description
1.0	Course of Action data model Version 1.0
1.0.1	Course of Action data model Version 1.0.1
1.1	Course of Action data model Version 1.1
1.1.1	Course of Action data model Version 1.1.1

3.2 StructuredCOAType Class

The `StructuredCOAType` class enables the specification of an alternative actionable structured representation for the Course of Action potentially for automated consumption and implementation. The `StructuredCOAType` class is an abstract class and is intended to be extended via a subclass to enable the expression of any structured course of actions. STIX has provided support for passing proprietary or externally defined structured courses of action using the `GenericStructuredCOAType` class (see [STIX_EXT]).

3.3 ObjectiveType Class

The `ObjectiveType` class characterizes the results that this Course Of Action is intended to achieve.

Figure 3-2. UML diagram of the `ObjectiveType` class

14

The property table, which includes property descriptions and corresponds to the UML diagram given in Figure 3-2, is provided in Table 3-3.

Table 3-3. Properties of the `ObjectiveType` class

Name	Type	Multiplicity	Description
Description	`stixCommon:StructuredTextType`	0..1	The `Description` property captures a textual description of the objective of this Course of Action. Any length is permitted. Optional formatting is supported via the `structuring_format` property of the `StructuredTextType` class.
Short_Description	`stixCommon:StructuredTextType`	0..1	The `Short_Description` property captures a short textual description of the objective of this Course of Action. This property is secondary and should only be used if the `Description` property is already populated and another, shorter description is available.
Applicability_Confidence	`stixCommon:ConfidenceType`	0..1	The `Applicability_Confidence` property characterizes the level of confidence in the asserted applicability of the suggested Course of Action for its targeted objective.

3.4   RelatedCOAsType Class

The `RelatedCOAsType` class specifies a set of one or more other Course of Actions asserted to be related to this Course of Action and therefore is a self-referential relationship. It extends the `GenericRelationshipListType` superclass defined in the STIX Common data model, which specifics the scope (whether the elements of the set are related individually or as a group).

The UML diagram corresponding to the `RelatedCOAsType` class is shown in Figure 3-3.
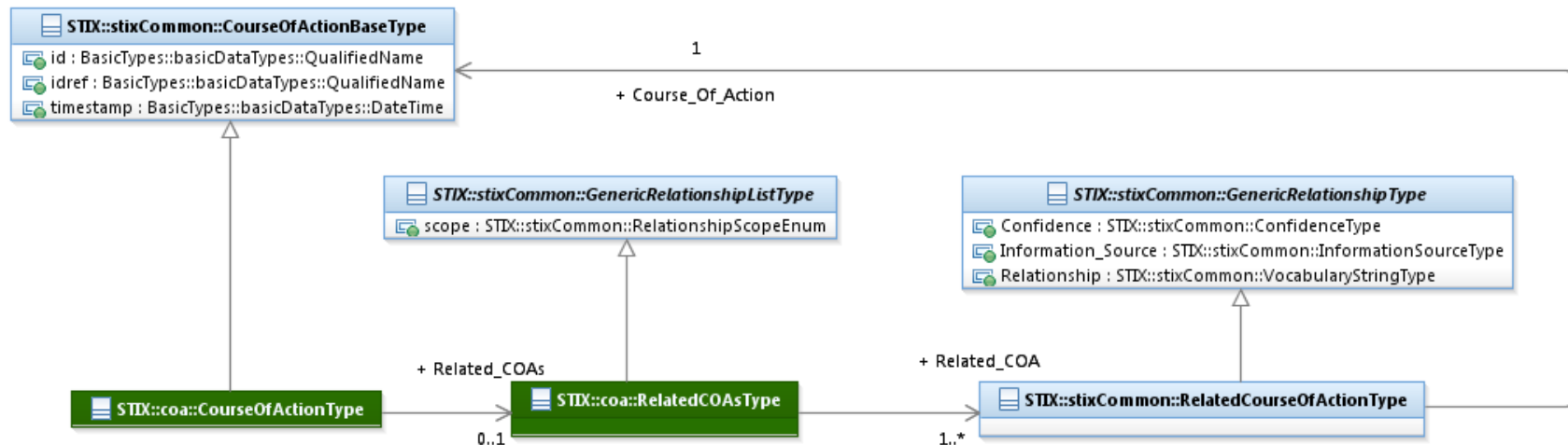
Figure 3-3. UML diagram of the `RelatedCOAsType` class

The property table given in Table 3-4 corresponds to the UML diagram shown in Figure 3-3.

Table 3-4. Properties of the `RelatedCOAsType` class

Name	Type	Multiplicity	Description
Related_Course of Action	`stixCommon: RelatedCourseOfActionType`	1..*	The `Related_COA` property specifies another Course of Action associated with this Course of Action and characterizes the relationship between the Courses of Action by capturing information such as the level of confidence that the Courses of Actions are related, the source of the relationship information, and type of the relationship. A relationship between Courses of Action may represent assertions of general associativity or different versions of the same Course of Action.

References

References made in this document are listed below.

[CybOX_{COR}]   CybOX^{TM} Core Specification (*not yet available*).

[REL]   STIX^{TM} Course of Action Model as implement in XSD
              https://stix.mitre.org/language/version4.1/xxx_schema.xsd

[RFC2119]   RFC 2119 – Key words for use in RFCs to Indicate Requirement Levels
              http://www.ietf.org/rfc/rfc2119.txt

[STIX]   STIX^{TM} Web Site
              https://stix.mitre.org

[STIX-SPECS] STIX^{TM} Project Github Site
              http://github.com/STIXProject/specifications

[STIX_O]   STIX^{TM} 1.1.1 Specification Overview
              http://stix.mitre.org/about/documents/XXXX.pdf

[TOU]   Terms of Use
              http://stix.mitre.org/about/termsofuse.html