ONTOLOGY FOR MEDIA CREATION
PART 7: RELATIONSHIPS

VERSION 2.8

Motion Picture Laboratories
December 5, 2024

Contents

This document is intended as a guide for companies developing or implementing products, solutions, or services for the future of media creation. No effort is made by Motion Picture Laboratories, Inc. to obligate any market participant to adhere to the recommendations in this document. Whether to adopt these recommendations in whole or in part is left to the discretion of individual market participants, using independent business judgment. Each MovieLabs member company shall decide independently the extent to which it will utilize, or require adherence to, these recommendations. All questions on member company adoption or implementation must be directed independently to each member company

1 Introduction

What is a relationship? At its simplest, it is a connection between two things. Relationships can also be very complex, influenced by qualifiers and circumstances. In the production process, in some ways everything is connected, directly or indirectly back to the script and forward to the finished creative work. One of the goals of the Ontology is to allow any component – Task, Asset, Participant, or something else – to make those connections. This often requires many hops though intervening connections.

This information is useful in several ways. At the most basic level, it shows dependencies: A scene depends on a location and some characters, a costume depends on a particular actor, a task requires certain inputs. This can make it easier to find the things you need to do a task because they are now connected. At a more complex level, it can reveal the consequences of change. Some of this is managed by automated systems today, but a great deal of it is handled, with much time and effort, by people.[1] The Ontology gives a path towards automating more of it, improving efficiency and reliability.

Some relationships are more or less linear, even if they are unexpected – they go from one thing to another. James Burke used this device in *Connections*, for example, moving from the development of large merchant ships by the Dutch in the 16th century, through better insurance rates at Lloyd's for ships with well-tarred hulls, to the development of coal tar in Scotland to meet this need, the discovery of ammonia and the search for synthetic quinine, to the development of artificial dyes, leading to significant advances in chemistry and on to synthetic fertilizer and improved food production.[2]

But impressive as it is, this linearity is a bit of lie, or at least a lie by omission. Every element in that chain has other antecedents and other consequences, which the writer chose to ignore for improved clarity and a bigger "Wow" factor.[3] Multiple paths are integral to relationships in the production process – it is a long and winding road from the Script to the finished creative work, with connections that split off then re-integrate.

This other extreme from the Burkean linearity is much better expressed by Douglas Adams, who calls it "the interconnectedness of all things"[4] or by John Donne with "No man is an Island, entire of itself; every man is a peece of the Continent, a part of the main."[5] This is more realistic, and it is the job of the Ontology to make it clear what the relationships are so that people and applications can make use of whichever of them they need.

1.1 Scope

The other parts of the Ontology cover relationships at a very high level, usually as attributes of particular elements (context components, assets, etc.), describing only the target of the relationship. This

[1] "…like a spider in the center of its web, but that web has a thousand radiations, and he knows well every quiver of each of them" – Sherlock Holmes in *The Final Problem*

[2] *Connections* (1978), episode 7, "The Long Chain," https://www.imdb.com/title/tt0818340/

[3] For example, what about the connection between the larger merchant ships and the increased availability of coffee in London? Lloyd's was the original coffee house where insurance was underwritten, rather than staring as an insurance venture.

[4] In *Dirk Gently's Holistic Detective Agency* (1987)

[5] *Devotions Upon Emergent Occasions*, Meditation XVII (1623)

document gives those relationships formal names. It also provides the inverses of the relationships, which are not generally present in the other documents and adds a few useful relationships that are implied by the other documents but not specifically mentioned.

Note, this document does not stand on its own. The nuances of many of the relationships are defined and explained with the concept to which they apply in the other parts of the Ontology.

1.2 Appropriate Granularity

All relationships can be refined as needed. For example, a Task may want to distinguish its inputs beyond the basic “input” relationship. This document provides some of these finer-grained relationships and will extend that set based on feedback from implementations.

1.3 Notational Conventions

In documents generally:

- The definition of a term included in the Dictionary is in bold, followed by the definition, e.g., **Creative Work:** A uniquely identified production.
- When a defined term is used in the text of a document, it is capitalized, for example in “The Production Scene is usually derived from a numbered scene in the Script,” Production Scene and Script are defined in the Ontology. (Note, a word that is part of defined term may sometimes be capitalized by itself as a shorthand, e.g., “Scene” may be used to indicate “Narrative or Production Scene.”)
- References to other Ontology Documents are in ***bold italic***, e.g., ***Part 3: Assets*** or ***Part 3A: Camera Metadata.***

For Sample Attributes in the concept documents:

- If a data field or attribute is formally defined in this ontology or a connected ontology, it is italicized, e.g., *Setup* as an attribute refers to a defined concept.
- Attribute […] indicates an attribute can appear more than once, e.g., *Identifier* […]
- →Thing means that an attribute is expressed as a relationship to a Thing, e.g., →*Script*
- A combination of the two indicates that the concept can have relationships to a set of things, e.g., →Components […]
- Many elements of the Ontology have a Context element. (See **Part 2: Context**.) Relationships declared in the Context are implied to have the item to which the Context is attached as their starting point. For example, Narrative Location→Context→Narrative Scene

  Contextual relationships that are especially important to the concept being defined are given in the sample attributes tables as C→Thing or C→Thing […] as appropriate. These relationships can just as well be on the object that has the Context. For example, if Narrative Location has “C→Narrative Scene” as an attribute, it is ok to have the relationship directly on the Narrative

Location or in its Context, e.g. Narrative Location→Narrative Scene or Narrative Location→Context→Narrative Scene.

Some implementations (e.g. RDF) place these Context-related relationships directly on the class as well as allowing them in Context, and others (e.g. JSON) only place these Context-related relationships in a Context. Relationships that are part of the class itself – not contextual – are always placed on the class.

2 Concepts and Terms

2.1 Relationships

Relationship: Describes and defines the connections between elements of the Ontology, such as Assets, Tasks, Participants, and Contexts.

Relationships are normally bidirectional – relationships almost always have a direction and an inverse. For instance, the inverse of “features” is “featuresIn”, as in NarrativeScene -> features -> NarrativeObject and its inverse, NarrativeObject -> featuresIn -> NarrativeScene.

In these cases, it is reasonable to infer that if one relationship is stated the other is true, which can be useful if an application only receives information on one direction of the relationship. This is not to be confused with a symmetric relationship, which is the same in both directions.

Relationships have a domain, the thing at which the Relationship starts, and a range, the thing to which the relationship connects. These are sometimes called “source” and “target,” respectively.

Relationships can be refined by subclassing. See below under “Naming Relationships.”

Notes:

The sections are arranged as follows:

- A Section on how relationships are named
- A section with “simple” relationships, organized by the part of the Ontology in which the relationship is defined (Context, Asset, etc.)
- A section on reified relationships

Some relationships have a very broad range, which is described in the Notes column.

The representation of arrays of relationships is implementation dependent. For example, an RDF implementation can just allow multiple instances of a particular relationships, while a JSON implementation might use an actual array of relationships.

2.1.1 How to read the tables

Most relationships are bidirectional, and both directions are documented in the same row of the table in which the relationship appears. The range of the original relationship is the domain of the inverse relationship, and this is indicated in each table’s two header rows. Arrows are included to indicate the direction of a relationship, going from its domain to its range.

As an example, the non-header row of this table should be read as follows:

- Narrative Scene is the domain of “features”, and its range is a Narrative Object
- The domain of “featuresIn” is a Narrative Object, and its range is a Narrative Scene.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Narrative Scene	features→	← featuresIn	*Narrative Object*	

2.2 Naming Relationships

OMC is as much about the relationships between things as it is about the thing themselves and it follows a set of conventions for their names. In the remaining sections, it will be noted when a relationship does not follow the expected pattern; this can result from problems of language or connotation, but is uncommon.

Relationships can be subclassed. For example, when it is known that the target of the "features -> Narrative Object" relationship is a Narrative Prop, the relationship can be "features.NarrativeProp".

Standard Basic Relationships

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Narrative Scene	features→	← featuresIn	*Narrative Object, Special Effect, Visual Effect, Special Action*	"features" connects Narrative Scenes to other narrative elements. It can be subclassed (see below for examples.) However, "featuresIn" is generally not subclassed, because A narrative Object and its subclasses are always featured in a Narrative scene.
Character	needs→	← neededBy	*Narrative Object, Special Effect, Visual Effect, Special Action*	"needs" connects a Character to other narrative elements. It can be subclassed. "neededBy" is not generally subclassed.
Production Scene	uses→	← usedIn	*Production Object, Portrayal, Depiction*	"usedIn" connects Production Scenes to production elements. It can be subclassed, e.g. to indicate that Production Set Dressing is

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
				usedIn.ProductionSet and usedIn.ProductionScene It can also be used to connect Production Objects to Production Sets.
Production Scene	representedBy→	← represents	*Asset*	This covers the realized performance of the Production Scene (a capture) or the creation of it by some other means (e.g. the results of CG rendering.) This is distinct from "Representation" (a type of Version.)
	has…→	← for…		"has…" and "for…" indicate that the relationship should or can be subclassed based on the type and its domain. These are used for intrinsic properties. There are some specialized named relationships where "for…" is used as the inverse.
Reified Objects	(special)→	← in…	(any*)*	Reified relationships (see below) use distinct names when pointing to the two elements of the reification; those two elements relate back to the reification with the "in…" relationship, which can be subclassed.
(any)	related…	related…		Anything in the Ontology can be related to anything else. This can be useful when carefully managed,

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
				and confusing when allowed to run wild. Several classes explicitly support a subclass of the related relationship.

2.3 Simple relationships

NOTE: This table of relationships is complete through OMC 2.1. A future version of this document will include relationships added since then. New relationships are, of course, covered in the documents that define them even though they are not yet included in these tables. Some notable additions are in ***Part 9: Utilities*** for Composition, and in ***Part 2: Context***, ***Part 3C: Audio***, ***Part 3D: CG Assets***, and ***Part 3F: Images*** for their new features, many of which use Compositions.

2.3.1 Media Creation Context

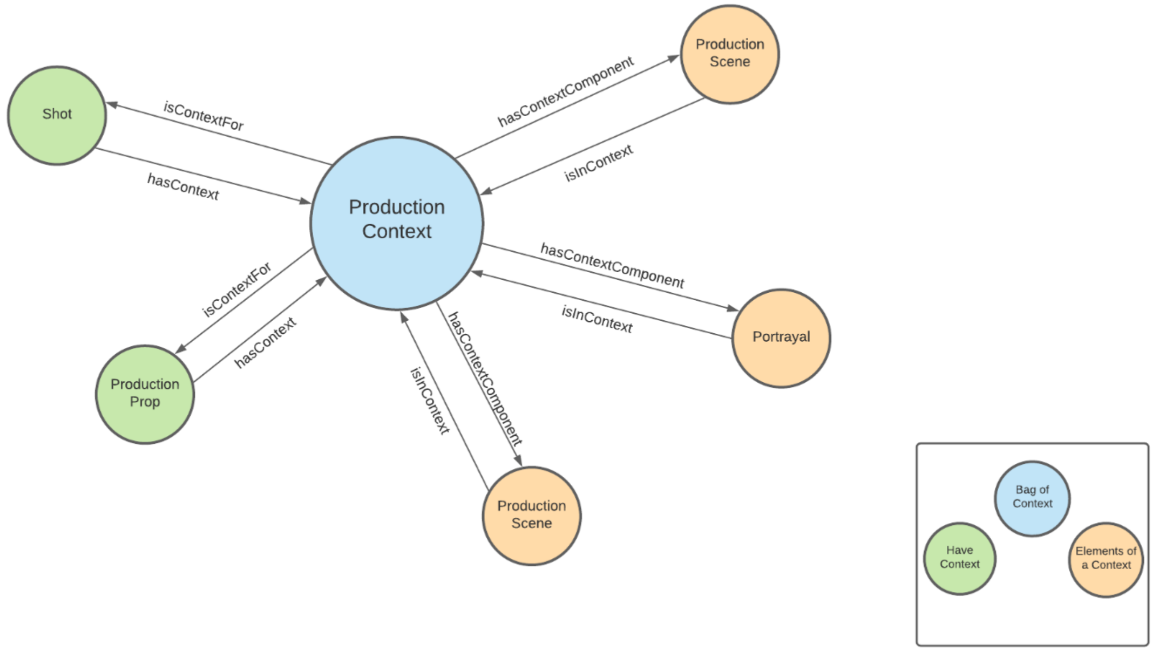
Media Creation Context and its subclasses (Narrative Context and Production Context) are special. Almost anything can be part of a context, and almost all elements of the Ontology can have a relationship to a context.

A Media Creation Context is the most general form of a context. The Ontology defines two subclasses for it, Narrative Context and Production Context (see ***Part 2: Context***) that are useful at the topmost level of production workflows, but we expect that individual parts of the production process will define their own types of Context, based on the Media Creation Context. For example, a VFX task may take a Production Context, and possibly a Narrative Context as its starting point, and then create a new Context that depends on the details of the process.

Managing relationships to and from contexts may result in circuitous circular references, and we expect that best practices for allowing and disallowing that will emerge.

Contexts themselves can have their own relationships, of course, e.g. “related.Context.”

The following diagram shows both kinds of relationships for a Production Context – its components and things that have a relationship to a context. A single Production Context is used to provide information for a particular shot and for a Production Prop in two particular scenes.



Context Components

The contents of a context depend on individual workflows. In general, a Narrative Context should contain only narrative items and a Production Context should contain production items, but this is not a hard and fast rule. Individual implementations can apply any restrictions needed to support a particular workflow, such as requiring a specific component or disallowing all but a small set of components; this can be done at the schema level or as runtime checks.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	**Note that these relationships are independent top-level relationships, not subclasses of "has" and "for".**
Media Creation Context	hasContextComponent →	← in.Context		Applies to both narrative and production contexts. This can be used to find anything in a Context.
Media Creation Context	related.Context →	← related.Context	*Media Creation Context*	Applies to both narrative and production contexts
Narrative Context	hasContextComponent →	← in.Context		The components of a Narrative Context are generally narrative

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	**Note that these relationships are independent top-level relationships, not subclasses of "has" and "for".**
				elements, rather than production elements
Production Context	hasContextComponent →	← in.Context		Production Contexts often have a mix of Narrative and Production elements.

Using Contexts

hasContext and its subclasses are defined only by the range (a context.) The domain is not restricted since we want to be able to attach a context to most things. It is included only here, rather than adding hasContext to almost every item in the Ontology.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	**"hasContext" and its two subclasses are independent top-level relationships.** **This uses "for" because the thing to which Context is attached is an intrinsic property of the Context. "for" can be subclassed to reflect its range (the domain of "hasContext", etc.**
	hasContext →	← for…	*Media Creation Context*	The domain is anything that can have a Context.
	hasProductionContext →	← for…	*Production Context*	Subclass of hasContext.
	hasNarrativeContext →	← for…	*Narrative Context*	Subclass of hasContext.

2.3.2 Characters and Narrative and Production Objects

These relationships all relate to the use of the various elements, rather than to their intrinsic nature. As noted above, that means they can exist on a Context or on something that matches the Range type. Implementations should be consistent about where they are placed, with a preference for placing them in a Context.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Character	featuresIn→	←features.Character	*Narrative Scene*	See Portrayal below for a Character's relationships to a Portrayal.
Narrative Object	neededBy.Character →	← needs.NarrativeObject	*Character*	
Narrative Object	featuresIn →	← features.NarrativeObject	*Narrative Scene*	For scheduling and budgeting it can be important to have props allocated to scenes. This relationship makes that
Narrative Prop	neededBy.Character →	← needs.NarrativeProp	*Character*	
Narrative Prop	featuresIn.NarrativeScene →	← features.NarrativeProp	*Narrative Scene*	For scheduling and budgeting it can be important to have props allocated to scenes.
Production Prop	usedIn.Portrayal →	← uses.ProductionProp	*Portrayal*	Generally, it is better to connect a Portrayal to a Depiction rather than directly to a Production Prop. See Depiction in Reified Relationships for how to do that.
Production Prop	usedIn.ProductionScene →	← uses.ProductionProp	*Production Scene*	Generally, it is better to connect a Production Scene to a Depiction rather than directly to a Production Prop. See Depiction in Reified Relationships for how to do that.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Narrative Set Dressing	featuresIn.NarrativeScene →	← features.NarrativeSetDressing	*Narrative Scene*	
Production Set Dressing	usedIn.ProductionSet →	← uses.ProductinSetDressing	*Production Set*	
Production Set Dressing	usedIn.ProductionScene →	← uses.ProductionScene	*Production Scene*	
Narrative Greenery	featuresIn.NarrativeScene →	← features.NarrativeGreenery	*Narrative Scene*	
Narrative Greenery	neededBy.Character →	← needs.NarrativeGreenery	*Character*	
Production Greenery	usedIn.ProductionSet →	←uses.ProductionGreenery	*Production Set*	
Production Greenery	usedIn.ProductionScene →	← uses. ProductionGreenery	*Production Scene*	
Production Greenery	usedIn.Portrayal →	← uses. ProductionGreenery	*Portrayal*	
Narrative Vehicle	neededBy.Character →	← needs.NarrativeVehicle	*Character*	
Narrative Vehicle	featuresIn.NarrativeScene →	← features.Vehicle	*Narrative Scene*	
Production Vehicle	usedIn.Portrayal →	← uses.ProductionVehicle	*Portrayal*	See Depiction below for a Production Vehicle's relationship to a Depiction.
Production Vehicle	usedIn.ProductionScene →	← uses.ProductionVehicle	*Production Scene*	
Narrative Wardrobe	neededBy.Character →	← needs.Wardrobe	*Character*	
Narrative Wardrobe	featuresIn.NarrativeScene →	← features.Wardrobe	*Narrative Scene*	

Domain	Relationship →		Range	Notes
Range		**← Relationship**	**Domain**	
Costume	usedIn.Portrayal →	← uses.Costume	*Portrayal*	
Costume	usedIn.ProductionScene →	← uses.Costume	*Production Scene*	
Narrative Hair	neededBy.Character →	← needs.Hair	*Character*	
Narrative Hair	featuresIn.NarrativeScene →	← features.Hair	*Narrative Scene*	
Production Hair	usedIn.Portrayal →	← uses.Hair	*Portrayal*	
Production Hair	usedIn.ProductionScene →	← uses.Hair	*Production Scene*	
Narrative Makeup	neededBy.Character →	← needs.Makeup	*Character*	
Narrative Makeup	featuresIn.NarrativeScene →	← features.Makeup	*Narrative Scene*	
Production Makeup	usedIn.Portrayal →	← uses.Makeup	*Portrayal*	
Production Makeup	usedIn.ProductionScene →	← uses.Makeup	*Production Scene*	
Narrative Prosthetics	neededBy.Character→	← needs.Prosthetics	*Character*	
Narrative Prosthetics	featuresIn.NarrativeScene →	← features.Prosthetics	*Narrative Scene*	
Production Prosthetics	usedIn.Portrayal →	← uses.Prosthetics	*Portrayal*	
Production Prosthetics	usedIn.ProductionScene →	← uses.Prosthetics	*Production Scene*	

2.3.3 Audio

Unless otherwise noted, these can all exist on a Context or on something that matches the Range type.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Narrative Sound Effect	featuresIn.NarrativeScene →	←features. NarrativeSoundEffect	*Narrative Scene*	
Narrative Music	featuresIn.NarrativeScene →	←features. NarrativeMusic	*Narrative Scene*	

2.3.4 Supporting Activities and Special Actions

Unless otherwise noted, these can all exist in a Context or on something that matches the Range type.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Special Effect	featuresIn.NarrativeScene →	←features.SpecialEffect	*Narrative Scene*	
Special Effect	neededBy.Character →	←needs.SpecialEffect	*Character*	
Visual Effect	featuresIn.NarrativeScene →	←features.VisualEffect	*Narrative Scene*	
Visual Effect	neededBy.Character →	←needs.VisualEffect	*Character*	
Stunt	featuresIn. NarrativeScene →	←features.Stunt	*Narrative Scene*	
Stunt	neededBy.Character →	←needs.Stunt	*Character*	
Fight	featuresIn. NarrativeScene →	←features.Fight	*Narrative Scene*	
Fight	neededBy.Character →	←needs.Fight	*Character*	
Choreography	featuresIn. NarrativeScene →	←features.Choreography	*Narrative Scene*	
Choreograpy	neededBy.Character →	←needs.Choreography	*Character*	
Marine	featuresIn. NarrativeScene →	←features.Marine	*Narrative Scene*	
Marine	neededBy.Character →	←needs.Marine	*Character*	
Aerial	featuresIn. NarrativeScene →	←features.Aerial	*Narrative Scene*	

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Aerial	neededBy.Character →	←needs.Aerial	*Character*	
Motion Capture	featuresIn. NarrativeScene →	←features.MotionCapture	*Narrative Scene*	
Motion Capture	neededBy.Character →	←needs.MotionCapture	*Character*	

2.3.5 Locations and Sets

Unless otherwise noted, these can all exist on a Context or on something that matches the Range type.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Narrative Location	has.Location →	←for.NarrativeLocation	*Location*	Location is an intrinsic property of the Narrative Location.
Narrative Location	featuresIn.NarrativeScene→	←features.NarrativeLocation	*Narrative Scene*	
Production Location	has.Location →	←for.ProductionLocation	*Location*	Location is an intrinsic property of Production Location.
Production Location	usedIn.ProductionScene →	←uses.ProductionLocation	*Production Scene*	
Production Set	usedIn.ProductionLocation →	← uses.ProductionSet	*Production Location*	
Production Set	usedIn.ProductionScene →	←uses.ProductionSet	*Production Scene*	

2.3.6 Scenes, Shots, and Sequences

Relationships of Scenes to Locations are covered above.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Narrative Scene	for.CreativeWork →	← has.NarrativeScene	*Creative Work*	
Narrative Scene	for.Script →	← has.NarrativeScene	*Script*	

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Production Scene	for.NarrativeScene →	←has.ProductionScene	*Narrative Scene*	In this case, “for” and “has” connect the Narrative and Production worlds.
Production Scene	related.ProductionScene →	← related.ProductionScene	*Production Scene*	This relationship is completely symmetrical, not just bidirectional.
Slate	has.CameraUnit →	← for.Slate	*Camera Unit*	A Camera Unit is a Participant
Slate	for.ProductionScene →	← has.Slate	*Production Scene*	
Slate	for.Asset →	← has.Slate	*Asset*	This is used to connect any Slate-specific information, e.g., on-set data, to a particular take
Slate	for.Capture→	← has.Slate	*Capture*	This is used to connect any the Slate to a Capture.
Slate	for.CameraMetadata →	← has.Slate	*Camera Metadata*	
Slate	has.Camera →	←for.Slate	*Camera*	
Slate	has.Director →	←for.Slate	*Participant*	This can be often be found through a chain of relationships back to the Creative Work, but some on-set practices record it as part of the Slate
Slate	for.CreativeWork →	← has.Slate	*Creative Work*	This is not necessary if the connections from Slate to Production Scene to Creative Work are maintained, but is included because some on-set practices record it as part of the Slate.
Capture	has.Slate →	← for.Capture	*Slate*	
Capture	represents.ProductionScene →	← representedBy.Capture	*Production Scene*	

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Shot	has.Slate →	←for.Shot	*Slate*	See **Part 2: Context** s.v. “Shot” for how to use Shot. The Shot’s Structural Characteristics represent the source media for the Shot.
Sequence	has.SCD →	← for.Sequence	*Sequence Chronology Descriptor*	This applies to all subclasses of Sequence.
Sequence Chronology Descriptor	has.SCDComponent →	← for.SCD	*SCD Component*	
SCD Component	has.Shot→	← for.SCDComponent	*Shot*	

2.3.7 Other Context Elements

Unless otherwise noted, these can all exist on a Context or on something that matches the Range type.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Script	for.CreativeWork →	←has.Script	*Creative Work*	
Script	has.Screenwriter →	←for.Script	*Participant*	
Storyboard	for.NarrativeScene →	← has.StoryBoard	*Narrative Scene*	Also applies to Animated Storyboard
Concept Art	→ for…	←has.ConceptArt		The range of Concept Art -> for is usually a narrative element but can also be a Script Concept Art can also be included in a Concept, which is a reification of this relationship.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Creative Reference Material	→ for…	← has.CreativeReferenceMaterial		Creative Reference material is often related to an entire production or entire sections of a production, in which case it is simpler to add it as a context element rather than use relationships to a plethora of narrative or production elements. This use of "for" is generally subclassed.
Technical Reference Material	→ for…	← has.TechnicalReferenceMaterial		This usually has a relationship to a Production Scene or other production element, such as a Production Set or a Slate. This use of "for" is generally subclassed.

2.3.8 Asset

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Asset	has.AssetStructuralCharacteristic →	← for.Asset	*Asset Structural Class*	
Asset	→ has.AssetFunctionalCharacteristic	← for.Asset	*Asset Functional Class*	
Asset	→ has.AssetGroup	← for.AssetGroup	*Asset Group*	
Asset Structural Class	→ uses.Carrier	← usedBy.AssetStructuralClass		Range can be an Asset or a piece of Infrastructure
Asset Group	→ member	← memberOf	*Asset*	

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Image Sequence	→ has.Image	← for.ImageSequence	*Image*	Image Sequence is an Asset Structural type, not an Asset Group.

2.3.9 Asset Versions and Provenance

Asset Versions

Versions follow a somewhat different pattern from other relationships. We are investigating ways of making them more conformant in a future release, and will keep these for backwards compatibility if they change.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Asset	→hasVersionInfo	←forAsset	*Version Info*	
Version Info	→ versionOf	← hasVersion	*Asset*	
Asset	→ hasRevisionInfo	←forAsset	*Revision Info*	
Revision Info	→ revisionOf	← hasRevision	*Asset*	
Asset	→ hasVariantInfo	← forAsset	*Revision Info*	
Variant Info	→ variantOf	← hasVariant	*Asset*	
Asset	→ hasDerivationInfo	← forAsset	*Derivation Info*	
Derivation Info	→ derivationOf	← hasDerivation	*Asset*	
Asset	→ hasRepresentationInfo	← forAsset	*Representation Info*	
Representation Info	→ representationOf	← hasRepresentation	*Asset*	
Asset	→ hasAlternativeInfo	← forAsset	*Alternative Info*	
AlternativeInfo	→ alternativeOf	← hasAlternative	*Asset*	

Notes:

> All of the Version Info types except for Alternative can be anchored with Asset Structural Characteristics, for example Asset Structural Characteristics →RevisionInfo → Asset Structural Characteristics. See the examples in ***Part 3B: Versions***

Provenance

Provenance can be applied to Assets or to Version Info and its subtypes. Deciding where to put it is up to individual workflow practices.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Asset	→has.Provenance	←for…	*Provenance*	
Asset Structural Characteristics	→ has.Provenance	← for…	*Provenance*	
Version Info	→ has.Provenance	← for.VersionInfo	*Provenance*	Includes all subtype of Version Info
Provenance	→ createdBy	← created	*Participant*	
Origin	→ hasOrigin	← isOriginFor		A relationship to something with an identifier, or a URL, or some string explaining the origin of the thing. In general, this will be set only at the beginning of an Asset chain.

2.3.10 Participant

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Participant	has. ParticipantStructuralCharacteristic →	← for.Participant	*Participant Structural Class*	
Participant	has. ParticipantFunctionalCharacteristic →	← for.Participant	*Participant Functional Class*	
Participant	has.ParticipantGroup →	← for.Participant	*Participant Group*	

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Participant Group	member →	← memberOf	*Participant*	

2.3.11 Task

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Task	has.TaskStructuralCharacteristic →	← for.Task	*Task Structural Class*	Reserved for future use
Task	has.TaskFunctionalCharacteristic →	← for.Task	*Task Functional Class*	
Task	→ has.TaskGroup	← for.Task	*Task Group*	
Task	has.Scheduling →	← for.Task		The Ontology does not define a formal model for scheduling. We expect that one or more connected ontologies will do such a thing. This relationship is a placeholder for connecting scheduling data of any sort to a Task.
Task	input…. →	← for.Task	*Asset*	This can be subclassed to specify a particular type of input, e.g. “input.Sequence”, “input.Asset.ProductionProp”, or “input.Portrayal”
Task	output…. →	← for.Task	*Asset*	Can be subclassed the like “input”
Task	informs →	← informedBy	*Task*	
Task Group	member →	← memberOf	*Task*	

2.3.12 Creative Work

Domain	Relationship →		Range	Notes
Range		**← Relationship**	**Domain**	
Creative Work	basedOn →	← basisFor		The range for this can include other Creative Works, as well as tings defined in other ontologies, such as books.
Creative Work	→ inspiredBy	← inspires		The range for this can include other Creative Works, as well as things defined in other ontologies, such as books.
Creative Work	→ has.Contribution	← for.CreativeWork	*Contribution*	
Creative Work	→ has.Portrayal	← for.CreativeWork	*Portrayal*	Portrayal should only be a portrayal by a Person. This is used when full Contribution records are not kept.
Creative Work	→ has.ProductionCompany	← for.CreativeWork	*Participant*	Participant is an Organization
Creative Work	→related.CreativeWork	← related.CreativeWork	*Creative Work*	related is symmetrical – it is the same in both directions.

2.4 Reified Relationships

In some circumstances, a relationship itself is an object of interest. This is done by creating a new thing that combines the source and target and target of the relationship with an expression of the original relationship into a new object, potentially along with other information. This process is called *reification*.[6]

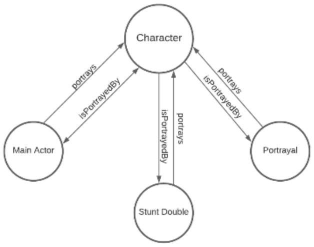
The Ontology reifies relationships when the relationship itself is of interest to the production process and needs to be communicated as a thing on its own. As an example, a character can be portrayed by multiple actors (main actor and stunt doubles, for example) or even by computer-generated character. A scene in the movie may use none, one, or all of them. The obvious simple relationship between

[6] "Reification" is "making a thing," ultimately from Latin *res*, "thing" or "matter," from which English also gets "republic" (*res publica*, the public matter) and the word "real" (from *realis*, the adjective of *res*.) "Reification" also has a specific meaning in Marxist philosophy.

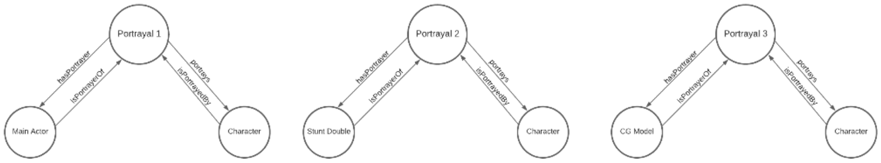
character and actor (or other portrayer) is “Character has.Portrayal Actor” but with that there is no easy way to add only relevant portrayals of the character to a Context.

Reifying this relationship retains the character to portrayer links indirectly, in this example through a new object called a portrayal. The portrayal can contain other information as well, and all of the relationships to and from the human and algorithmic portrayers still exist, though they are not shown in this diagram.

Pre-reification:



Post-reification:



The section heading here refer to the document in which the concepts are found.

NOTE: The following tables of relationships are complete through OMC 2.1. A future version of this document will include relationships added since then. New relationships are, of course, covered in the documents that define them even though they are not yet included in these tables.

2.4.1 Context

The details of Concept, Depiction, and Portrayal are covered in ***Part 2: Context***.

Domain	**Relationship →**		**Range**	**Notes**
Range		**← Relationship**	**Domain**	
Concept	subject→	← in.Concept		The range of "subject" is an element of the narrative, e.g., a Narrative Prop or a Character.
Concept	idea→	← in.Concept	*Asset*	The Asset is usually a piece of Concept Art, but could be any Asset, such as an Image Sequence (e.g., for an animated storyboard), a Moving Image, or Audio.
Depiction	depicts…. →	← in.Depiction	*Narrative Object*	The range of depicts is a Narrative Object, and can be subclassed accordingly.
Depiction	depicter…. →	← in.Depiction	*Production Object*	The range of depicts is a Production Object, and "depicter" can be subclassed accordingly.
Depiction	→usedIn.ProductionScene	← uses.Depiction	*Production Scene*	
Depiction	→usedIn.Portrayal	← uses.Depiction	*Portrayal*	
Portrayal	portrays →	←in.Portrayal	*Character*	A Portrayal is always of a Character
Portrayal	portrayer…→	← in.Portrayal		This can be a Participant or an Asset (e.g., a puppet, or a computer-constructed portrayer) and can be subclassed as needed.
Portrayal	→usedIn	← uses.Portrayal	*Production Scene*	

2.4.2 Participant

The relationship of a Participant to a Task is reified, since a Participant/Task pair is needed for some forms of security authorization. See ***Part 4: Participants***.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Work Unit	→ has.Participant	← in.WorkUnit	*Participant*	
Work Unit	→ has.Task	← in.WorkUnit	*Task*	

2.4.3 Creative Work

Contribution is reified for much the same reasons as Portrayal. One participant can be a contributor in multiple ways, such as being an actor and director, or portraying two characters. See ***Part 6: Creative Work*** for a fuller explanation of Contribution.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
Contribution	→ has.Contributor	← in.Contribution	*Participant*	
Contribution	→ has.Portrayal	← in.Contribution	*Portrayal*	

Appendix A External Definitions

These are terms defined elsewhere in the Production Ontology, included here for ease of reference.

Media Creation Context: Informs scope within the construction process of a Creative Work.

See **Part 2: Context**

Asset: A physical or digital object or collection of objects specific to the creation of the Creative Work.

See **Part 3: Assets**

Camera Metadata: Capture-specific details and information about the Camera itself.

See **Part 3A: Camera Metadata**

Participant: The entities (people, organizations, or services) that are responsible for the production of the Creative Work.

See **Part 4: Participants**

Task: A piece of work to be done and completed as a step in the production process.

See **Part 5: Tasks**

Creative Work: A uniquely identified production.

See **Part 6: Creative Works**

Relationship: Describes and defines the connections between elements of the Ontology, such as Assets, Tasks, Participants, and Contexts.

See **Part 7: Relationships**

Infrastructure: The underlying systems and framework required for the production of the Creative Work; it is generally not specific to a particular Creative Work.

See **Part 8: Infrastructure**

Utilities: Common data models and data structures used in multiple places and in multiple ways in a larger system.

See **Part 9: Utilities**

Identifier: An identifier uniquely identifies an entity within a particular scope.

See **Part 9: Utilities**