



ONTOLOGY FOR MEDIA CREATION

PART 7: RELATIONSHIPS

VERSION 1.0.1

Contents

1	Introduction	1
1.1	Scope	1
1.2	Appropriate Granularity.....	2
1.3	Notational Conventions	2
2	Concepts and Terms	3
2.1	Relationships.....	3
2.1.1	How to read the tables	3
2.2	Simple relationships.....	4
2.2.1	Media Creation Context.....	4
2.2.2	Other Context Elements	6
2.2.3	Asset	9
2.2.4	Participant.....	9
2.2.5	Task	10
2.2.6	Creative Work	10
2.3	Reified Relationships	11
2.3.1	Context.....	12
2.3.2	Participant.....	13
2.3.3	Creative Work	13
2.4	hasRelated....	13
2.4.1	hasRelatedAsset.....	14
2.4.2	hasRelatedParticipant.....	14
2.4.3	hasRelatedInfrastructure	15
Appendix A	External Definitions.....	16





© 2021 Motion Picture Laboratories, Inc.

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 through 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 Iland, intire of itselfe; every man is a peece of the Continent, a part of the maine.”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 starting 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 hasInput 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., Unstructured Data [...]
- →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 [...].

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 “Narrative Context hasComponent Character” is “Character isComponentOf Narrative Context.”⁶

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. `hasInputAssets` and `hasOutputAssets` can be subclassed for particular kinds of task. For instance, `hasVFXSequenceInput` might be a subclass of `hasInput` for a VFX Task.

Notes:

The sections are arranged as follows:

- 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
- A section on relationships derived from `hasRelatedAsset/Participant/etc.`

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

Some classes have an attribute called “Relationships,” which can include any relationship not covered in the other attributes; this attribute is not further addressed here.

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:

- Script is the domain of `hasAuthor`, and the range of `hasAuthor` is Participant

⁶ This is not to be confused with a symmetric relationship, which is the same in both directions.

- The domain of isAuthorOf is participant, and the range of isAuthorOf is Script.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Script</i>	hasAuthor →	← isAuthorOf	<i>Participant</i>	The Participant must be a person

2.2 Simple relationships

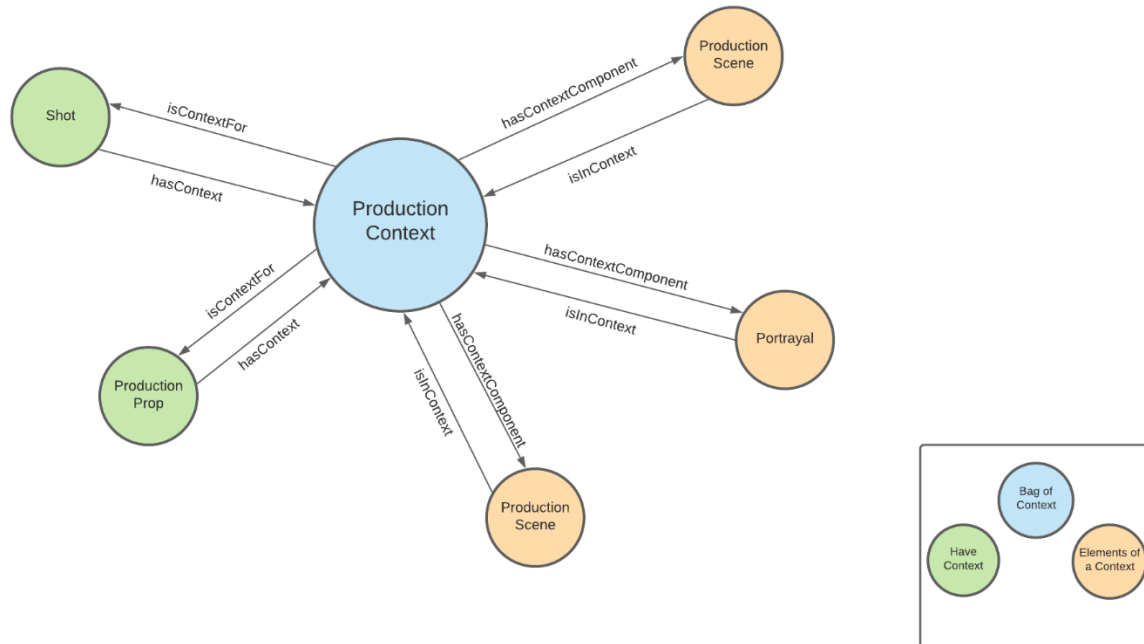
2.2.1 Media Creation Context

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.

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	
<i>Media Creation Context</i>	hasContextComponent →	← isInContext		Applies to both narrative and production contexts
<i>Media Creation Context</i>	hasRelatedContext →	← hasRelatedContext	<i>Media Creation Context</i>	Applies to both narrative and production contexts
<i>Narrative Context</i>	hasContextComponent →	← isInContext		The components of a Narrative Context are generally narrative elements, rather than production elements

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Production Context</i>	hasContextComponent →	← isInContext		The components of a Production Context are generally narrative elements, rather than 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 →	← isContextFor	<i>Media Creation Context</i>	The domain is anything that can have a Context.
	hasProductionContext →	← isProductionContextFor	<i>Production Context</i>	Subclass of hasContext/ isContextFor
	hasNarrativeContext →	← isNarrativeContextFor	<i>Narrative Context</i>	Subclass of hasContext/ isContextFor

2.2.2 Other Context Elements

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Narrative Location</i>	hasLocation →		<i>Location</i>	
<i>Production Location</i>	hasLocation →		<i>Location</i>	
<i>Narrative Prop</i>	isUsedBy →	← uses	<i>Character</i>	Other information about when and where the Narrative Prop is used is found in a Context.
<i>Narrative Prop</i>	isUsedIn →	← usesProp	<i>Narrative Scene</i>	For scheduling and budgeting it can be

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
				important to have props allocated to scenes. This relationship makes that simpler than going through a Context.
<i>Production Prop</i>	isUsedBy →	← uses	<i>Portrayal</i>	Other information about when and where the Production Prop is used is part of its Context.
<i>Set Dressing</i>	isUsedBy →	← uses	<i>Production Set</i>	
<i>Narrative Wardrobe</i>	isWardrobeFor →	← usesWardrobe	<i>Character</i>	
<i>Costume</i>	isCostumeFor →	← hasCostume	<i>Portrayal</i>	
<i>Narrative Scene</i>	isFromWork →	← hasNarrativeScene	<i>Creative Work</i>	
<i>Narrative Scene</i>	isFromScript →	← hasNarrativeScene	<i>Script</i>	
<i>Production Scene</i>	hasNarrativeScene →	← hasProductionScene	<i>Narrative Scene</i>	
<i>Production Scene</i>	hasRelatedScene →	← hasRelatedScene	<i>Production Scene</i>	This relationship is completely symmetrical, not just bidirectional.
<i>Slate</i>	isSlateFor →	← hasSlate	<i>Production Scene</i>	
<i>Slate</i>	isSlateFor →	← hasSlate	<i>Technical Reference Material</i>	
<i>Slate</i>	hasCreativeWork →		<i>Creative Work</i>	
<i>Capture</i>	hasSlate →	← isSlateFor	<i>Slate</i>	
<i>Capture</i>	hasProductionScene →	← hasCapture	<i>Production Scene</i>	
<i>Shot</i>	hasSource →	← isSourceForShot		See Part 2: Context s.v. “Shot” for the possible range.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Sequence</i>	hasSCD →	← isSCDFor	<i>Sequence Chronology Descriptor</i>	This applies to all subclasses of Sequence.
<i>Sequence</i>	hasSequenceElement →	← isSequenceElementOf	<i>Shot</i>	The range is usually a Shot, but it can be other things.
<i>Storyboard</i>	isStoryboardFor →	← hasStoryboard	<i>Narrative Scene</i>	
<i>Storyboard</i>	isStoryboardFor →	← hasStoryboard	<i>Production Scene</i>	
<i>Concept Art</i>	→ isConceptArtFor	← hasConceptArt		The range of isConceptArtFor is usually a narrative element. Concept Art can also be included in a Concept, which is a reification of this relationship.
<i>Creative Reference Material</i>	→ isCreativeReferenceMaterialFor	← hasCreativeReferenceMaterial		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.
<i>Technical Reference</i>	→ isTechnicalReferenceMaterialFor	← hasTechnicalReferenceMaterial		This usually has a relationship to a Production Scene or other production element, such as a Production Set

2.2.3 Asset

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Asset</i>	hasAssetStructuralCharacteristic →	← isAssetStructuralCharacteristicFor	<i>Asset Structural Class</i>	
<i>Asset</i>	→ hasAssetFunctionalCharacteristic	← isAssetFunctionalCharacteristicFor	<i>Asset Functional Class</i>	
<i>Asset</i>	→ hasAssetGroup	← isAssetGroupFor	<i>Asset Group</i>	
<i>Asset Structural Class</i>	→ hasCarrier	← isCarrierFor		Range can be an Asset or a piece of Infrastructure
<i>Asset Group</i>	→ hasAssetComponent	← isAssetComponentOf	<i>Asset</i>	
<i>Image Sequence</i>	→ hasImageComponent	← isImageComponentOf	<i>Image</i>	

2.2.4 Participant

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Participant</i>	hasParticipantStructuralCharacteristic →	← isParticipantStructuralCharacteristicFor	<i>Participant Structural Class</i>	
<i>Participant</i>	→ hasParticipantFunctionalCharacteristic	← isParticipantFunctionalCharacteristicFor	<i>Participant Functional Class</i>	
<i>Participant</i>	→ hasParticipantGroup	← isParticipantGroupFor	<i>Participant Group</i>	
<i>Participant Group</i>	→ hasParticipantComponent	← isParticipantComponentOf	<i>Participant</i>	

2.2.5 Task

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Task</i>	hasTaskStructuralCharacteristic →	← isTaskStructuralCharacteristicFor	<i>Task Structural Class</i>	Reserved for future use
<i>Task</i>	→ hasTaskFunctionalCharacteristic	← isTaskFunctionalCharacteristicFor	<i>Task Functional Class</i>	
<i>Task</i>	→ hasTaskGroup	← isTaskGroupFor	<i>Task Group</i>	
<i>Task</i>	→ hasScheduling	← isSchedulingFor		The Ontology does not fine 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.
<i>Task</i>	→ hasInputAsset	← isInputAssetFor	<i>Asset</i>	
<i>Task</i>	→ hasOutputAsset	← isOutputAssetOf	<i>Asset</i>	
<i>Task</i>	→ informs	← isInformedBy	<i>Task</i>	
<i>Task Group</i>	→ hasTaskComponent	← isTaskComponentOf	<i>Task</i>	

2.2.6 Creative Work

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Creative Work</i>	basedOn →	← providesBasisFor		The range for this can include other Creative Works, as well as things defined in other ontologies, such as books.
<i>Creative Work</i>	→ inspiredBy	← providesInspirationFor		The range for this can include other Creative Works, as well as things defined in other ontologies, such as books.
<i>Creative Work</i>	→ hasContribution	← isContributionTo	<i>Contribution</i>	

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Creative Work</i>	→ includesPortrayal		<i>Portrayal</i>	Portrayal should only be a portrayal by a Person.

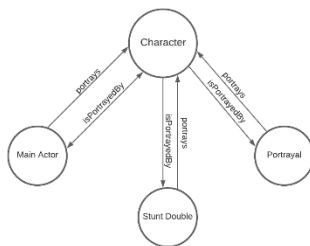
2.3 Reified Relationships

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*.⁷

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 character and portrayer is “character isPortrayedBy portrayer” but with that there is no easy way to add only relevant portrayals of the character to a context.

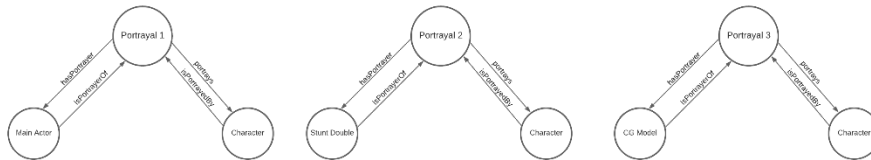
Reifying this relationship retains the character to portrayer links indirectly, in this example through a new object called a portrayal. The portrayal can contain other information as well, and all of the relationships to and from the human and algorithmic portrayers still exist, though they are not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

Pre-reification:



⁷ “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.

Post-reification:



2.3.1 Context

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

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Concept</i>	hasSubject →	← isSubjectOf		The range of hasSubject is an element of the narrative, e.g., a Narrative Prop or a Character.
<i>Concept</i>	hasIdea→	← isIdeaFor	<i>Asset</i>	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.
<i>Depiction</i>	depicts →	← hasDepiction		The range of depicts is narrative element.
<i>Depiction</i>	hasDepicter →	← isDepicterFor		The Depicter is usually a production element.
<i>Portrayal</i>	portrays →	← isPortrayedBy	<i>Character</i>	
<i>Portrayal</i>	hasPortrayer →	← isPortrayerOf		This can be a Participant or a computer-constructed portrayer, which will usually be an Asset.

2.3.2 Participant

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

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Role</i>	→ hasRoleParticipant	← isParticipantInRole	<i>Participant</i>	
<i>Role</i>	→ hasRoleTask	← isTaskInRole	<i>Task</i>	

2.3.3 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	
<i>Contribution</i>	→ hasContributor	← isContributorIn	<i>Participant</i>	
<i>Contribution</i>	→ hasContributionCharacter	← isContributionCharacterIn	<i>Character</i>	
<i>Creative Work</i>	isRelatedTo →	← isRelatedTo	<i>Creative Work</i>	An unreified isRelatedTo is symmetrical: the relationship is true in both directions. See Part 6: Creative Works for a reified version of this relationship, which will be included in future versions of this document.

2.4 hasRelated....

A production has to manage many thousands of relationships. Just as importantly, there are many different types of relationships, and managing those type is a significant problem. Many of these relationships are to Assets, Participants, and Tasks. It is impossible to enumerate all of them, and overwhelming to enumerate even the ones we know about now.

To make this easier to manage, we have several generic relationships: hasRelatedAsset, hasRelatedParticipant, and hasRelatedTask, which can be narrowed in meaning in a number of ways. In all cases, the range of the relationship can be narrowed as appropriate. For example,

- a system that allows subclassing of relationships can create the subclasses as desired;

- hasRelated... can be reified (see above), for example as a hasRelatedAsset class that contains a subtype (what kind of hasAssetRelationship it is, perhaps as an item of controlled vocabulary), and the target of the relationship;
- the specialized relationship can just be defined and declared on its own.

In all cases, the range of the relationship can be narrowed as appropriate. All of these methods have advantages and disadvantages, and the one chosen by an implementation will depend on the need of the applications using that implementation.

Some of this kind of relationship are so important (for reasons of clarity, intelligibility, or pervasive industry practice) that they are given in the tables above. In general, anything that has, say, Asset as its range can be done as a specialization of hasRelatedAsset. The following sections have all the other relationships defined in the Ontology that can be implemented as specializations of the hasRelated... relationships.

The format of the sections is the same as the relationships described in the rest of the document.

2.4.1 hasRelatedAsset

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Slate</i>	hasCameraMetadata →	← isCameraMetadataFor	<i>Camera Metadata</i>	Camera Metadata is an Asset

2.4.2 hasRelatedParticipant

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Script</i>	hasAuthor →	← isAuthorOf	<i>Participant</i>	The Participant must be a person
<i>Slate</i>	hasCameraUnit →	← isCameraUnitFor	<i>Participant</i>	The Participant should be a Camera Unit.
<i>Slate</i>	hasDirector →	← isDirectorFor	<i>Participant</i>	Having this explicitly on the Slate makes it easier to find, especially for parts of the workflow that may not have full access to a Participant list to search.
<i>Creative Work</i>	hasProductionCompany →	← isProductionCompanyFor	<i>Participant</i>	Participant must be an Organization

2.4.3 hasRelatedInfrastructure

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Slate</i>	hasCamera →	← isCameraFor	<i>Camera</i>	

2.4.4 hasRelatedLocation

In this case, the Ontology does not define any subclasses. This relationship serves as a base for subclassed relationships that support building hierarchies of Locations with appropriate granularity.

Domain	Relationship →		Range	Notes
Range		← Relationship	Domain	
<i>Location</i>	hasRelatedLocation →	← hasRelatedLocation	<i>Location</i>	

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**