A CREATIVE WORKS ONTOLOGY FOR THE FILM AND TELEVISION INDUSTRY

EXECUTIVE OVERVIEW

If data is a key to the future, the film and television industry has already embarked on a journey to unlock a better future. Every major media company is working intensively to integrate existing data systems and create innovative new systems to deliver more and better insights to decision makers. Those efforts are steadily improving enterprise-level data capabilities across the industry.

But as media companies add more and more data sources to fuel machine learning and other future applications to generate even deeper insights, the industry needs to do even more—not just better systems in every enterprise, but better systems and tools for the industry as a whole. That starts with core data infrastructure. There is a growing need for a standard, organizing framework to capture and surface the inherent relationships between works and other entities as part of that core data infrastructure.

This white paper describes one key component to help build the infrastructure of the future—a creative works ontology for film and television. An ontology provides a machine-to-machine framework for automating connections between the disparate data systems that populate the media world. It is core infrastructure and basic plumbing for a new world powered by data.

As such, it is not simply a new schema like the many others deployed across the industry to organize data. Creating a schema is a tactical choice for organizing data. Adopting an ontology is a strategic choice to promote fundamental infrastructure that helps the industry build compatible schemas. It serves to reduce duplication of effort both within and across enterprises. And as adoption increases across the industry among content producers, distribution platforms, and service providers, it also becomes a foundation for the development of other business applications in more rapid and scalable ways.

That is why MovieLabs and its member studios have collaborated to create an ontology that serves the film and television industry. This white paper describes the results of that effort, explaining the need for an ontology, its many potential use cases, and its key elements. It also attempts to answer some basic questions about the usefulness of an ontology versus other ways of organizing data, as well as the rationale for the technical choices made in designing this particular ontology for the film and television business.

The goal of the ontology and this white paper is to help enable the development of interoperable data systems and drive adoption of core infrastructure that can build a better and more competitive data future for the industry.

WHAT IS AN ONTOLOGY?

An ontology is a framework for defining the things, concepts and relationships that describe a domain of knowledge. It provides explicit definitions in machine-to-machine form that can be used to organize and connect data from multiple sources within the given domain. The ontology discussed in this white paper concentrates on the things, concepts and relationships commonly used to describe creative works in the film and television industry and is intended as a general resource for metadata creation, analytics, distribution, archiving, academic research and other use cases within the industry.

A successful ontology asks, "What data and connections do we care about?" The ontology then uses the answers to reduce complexity and add clarity. It focuses on the things we care about ("entities") and describes them using:

- Types: the kinds of entities (like movies, people, and locations),
- Properties: information that describes and distinguishes the entities (like titles for movies, names for people, and countries for locations),
- Relationships: how entities connect (e.g., a person directed a movie, a movie is set in a location).

Historically, the industry has spent more time describing individual works, and less time describing relationships among works and other entities. Connections between works, however, are becoming ever more important as the industry relies increasingly on analysis of large data sets spanning portfolios, marketing of families of films in the form of franchises and brands, and exploitation of content across multiple channels and platforms through film/TV crossovers, games, books, and theme park rides. An ontology focuses relatively more attention on the relationships between entities, not merely the properties of entities. This focus helps to enable the creation of systems and databases (like triple stores[1]) that incorporate standardized relationships as a core architectural component. An ontology also builds in numerous technical definitions and hooks—e.g., subclasses, ways of stating equivalence (or not)—that enable interaction between data systems and sources, improve applications relying on the ontology, and enable new capabilities as adoption expands across the industry.

The end result is an enabling model that defines both precise attributes of works and other entities and precise relationships among them, and also serves as a machine-to-machine framework for connecting data from different sources. Those characteristics make an ontology a valuable tool for data technologists and application developers whose job it is to automate the collection of data from

[1] See discussion of RDF triples below.

multiple sources and build working solutions that bridge data systems to create insights and power intuitive applications.

WHY THE FILM AND TELEVISION BUSINESS NEEDS AN ONTOLOGY

The film and television industry becomes more data-driven with each passing year. That trend is set to continue as the industry pulls in data from even more diverse distribution platforms and sources. The increasing focus on data highlights the need to connect the disparate data sources and disconnected data silos that populate the entertainment landscape.

Indeed, the industry often seems overpopulated with walled data gardens, incompatible data systems, inconsistent data standards, and frequently mundane obstacles to improvements in data competitiveness. Every major media organization expends significant resources overcoming those obstacles, integrating data sources and building data stores to serve different departments and purposes. That involves big commitments to data ingestion, mapping, integration, and normalization across numerous important data sources globally. Those efforts require the constant and detailed attention of data technologists, analysts, and application developers, making disparities between data systems more visible and more potentially problematic for mission-critical applications.

A standard ID like EIDR is designed to help overcome those data aggregation obstacles. It provides a common key for unlocking and linking data systems. The key acts as a property of a work and connects two records about the same work. It also connects and defines the relationships between different versions or manifestations of a work.

However, an ID by itself does not define the relationship between different works, e.g., whether one is a sequel or prequel of the other or whether both are part of the same fictional universe. Nor does an ID define the relationship between a work such as a film and a different entity like a song, book, location, character, theme park ride, or an item of merchandise such as an action figure.

A common industry ID is an excellent step that helps glue together records in different systems, but more is needed. An ontology goes beyond a common key to create a common framework for understanding the data inside a record. That framework helps enable integration of data from different systems through well-defined classes and subclasses of entities and their properties, as well as relationships among works and other entities. An ontology uses a standard ID to connect records in different data systems, but also constructs a translation map between the different data structures inside each record. In other words, it provides more and different glue for connecting a broader matrix of datapoints in different data frameworks.

It does all that in a machine-to-machine readable format that helps application developers and data technologists build cool and valuable things, while offering potentially dramatic reductions in time and resources devoted to melding data from the many sources becoming critical to industry success.

The industry—studios, distributors, analytics service providers and more—would be well-served to adopt such a capability. All indications are that the industry will continue to move more aggressively toward data-driven applications and decision-making. Application developers and data analysts will seek to learn more from more data sources and use data more effectively. As the business grows, studios will acquire more consumption data from more sources, requiring new and better tools to analyze that data for purposes of competitive analysis, affinity analysis, and marketing decision-making across franchises, universes, and channels. Studios and distributors will continue to deploy more powerful machine learning and AI tools that thrive on massive aggregations of data from multiple sources. The challenges an ontology addresses will only increase. Adoption and deployment of a common, machine-to-machine ontology will result in faster, more efficient integration of data, faster, more efficient development of applications to keep the industry competitive, and faster, more meaningful insights from data that includes connections surfaced by the ontology.

HOW THE INDUSTRY CAN USE A SHARED ONTOLOGY

An ontology is like plumbing. It is built into the walls and foundation of data structures. It provides both an enabling framework for building the data structure and standardized connectors for linking the data structure to other data systems and the broader data infrastructure of the industry. It essentially provides new and more powerful digital architecture for the future data requirements of the film and television business. That architecture becomes more powerful over time as adoption increases and industry participants both refine the common standard and share more compatible data with partners.

FOUNDATIONAL ENTERPRISE APPLICATIONS

Data technologists across the industry are building data management systems that require assumptions about entities, properties and relationships. Technologists borrow from industry standards when available and invent new data frameworks when necessary. A common ontology provides machine-to-machine definitions of entities, properties and relationships that serve as components for internal data structures and ready-made connectors to external data systems.

Those standardized components provide a foundation for data systems that can be more robust and flexible than custom, non-standard solutions. A common ontology has built-in hooks and connectors to enable adaptability and easier integration with other systems, e.g., a generic model for grouping of works that can map to many different models for specific types of groups. It incorporates industry definitions that support new and potentially powerful data structures—like triple stores that rely on pre-defined relationships in addition to properties, and graph databases that leverage those relationships in new and interesting ways. An ontology addresses data structuring questions with solutions already vetted and tested by others in the industry, allowing data technologists to focus on new extensions and innovations, not problems that have already been solved. (For example, the ontology has a pre-defined model for describing both the relationships between film works and the relationships between film works and other things such as books or games.) And since ontologies are designed to be more easily extensible than traditional schemas, data technologists can add granularity, definitions, or application extensions to support new use cases. (Examples include adding detailed consumption data or more granular character modeling into structures built into the current model and reserved for those purposes.)

A common ontology reduces duplication of effort both within and across enterprises. And importantly, once in place, it becomes an enterprise foundation for development of other business applications in a more rapid and scalable way.

DATA EXCHANGE AND WAREHOUSING

A primary application of a shared ontology is to enable the exchange of data with less reliance on many-to-many data mapping of systems and knowledge structures. Many-to-many mapping is time-consuming and expensive and a recurring constraint on the integration of new data sources. A common industry ontology makes it possible to map any data source or data sink once to the ontology, then more easily ingest data from any mapped source into any mapped sink. For a simple example, many entertainment databases map to Amazon IMDb or Box Office Mojo as common sources and transform that data into the unique format of an internal database; a single mapping of those sources to the ontology would allow any number of internal databases to take advantage of the data without repeating the source mapping for every application.

Data mappings can be shared openly among partners and industry participants to reduce unnecessary duplication of effort. Each new standardized mapping enables users to link more easily to other data systems, reducing the need for custom mapping between data sources and applications. For example, in addition to IMDb or Box Office Mojo, mappings could be created for other common industry data sources such as regional genre systems, ratings systems, rankings and commonly licensed commercial sources. Each new mapping could be used by multiple new data consumers (new companies, new internal databases, or new applications), who would need only to map to the ontology once to take advantage of data from multiple sources.

That benefit can be achieved both within an enterprise (that struggles with internal data silos) and between different enterprises with incompatible data systems. The end goal is to replace numerous variations of many-to-many custom mappings with a much smaller number of standardized one-to-many or many-to-one mappings. Multiplied across many applications and databases, the reduction in the volume of custom, many-to-many mappings can drive large savings in time and effort and reduce the costs of developing innovative new data-driven solutions.

SHARED DATA REPOSITORIES

The benefits of one-to-many mapping also makes a common ontology an apt framework for the creation and maintenance of shared industry data repositories. Contributors map once and then add data into a well-understood data structure that organizes data from multiple contributors, creating an aggregated data set that is greater than the contributions of any one participant.

Because the ontology is structured to include relationships and groupings, in addition to properties, and because the shared framework allows shared curation, the repository can also be broader and deeper than existing individual or shared data sets. For example, release dates for different categories of releases—named and structured differently and with different scope of coverage across multiple databases—can be mapped to a common ontology and combined to create a jointly curated collection of release dates covering more works and categories of releases across more territories than any of the contributing sources. Similarly, databases that focus largely on metadata for particular works, with only sparse data about the connections between works, can be combined to create a more fully developed set of relationships between works to identify common brands, franchises, and characters across portfolios. In turn, these can then be shared between participants and partners, augmenting and improving the internal databases of all contributing parties, as well as enabling new marketing and distribution opportunities on retailer platforms and multi-studio platforms like Movies Anywhere, UltraViolet, and others.

ANALYTICS & MARKETING APPLICATIONS

New augmented data sets created with a common ontology also enable more powerful analytics engines. Following the example described above, a dataset that aggregates release dates and relationship data from numerous sources offers the opportunity to cluster works for analysis using a broader set of commonly defined relationships, groupings and categories of release dates, e.g., affinity analysis and competitive assessments across franchises, brands and windows. Comparisons can be made across broader collections of aggregated data. Internal proprietary data can be combined with external data sources more quickly and efficiently to improve performance predictions for new titles and maximize direct-to-consumer marketing opportunities with customer targeting based on larger and more informative datasets. Machine learning algorithms can be written to find and analyze all available data mapped to a common concept in the ontology, making it easier to create the large volume of data necessary for machine learning applications and then turn massive aggregations of data from multiple sources into actionable intelligence[2].

[2] For an example of the kinds of things that can be done, see General Insights and Data Analysis and Research in https://www.bfi.org.uk/archive-collections/bfi-filmography-project-overview, a project that aggregated three data sources for archival and academic research.

CONSUMER APPLICATIONS

Larger shared datasets with common descriptors also provide the basis for more powerful consumer applications. An open ontology enables application developers to take greater advantage of the many existing public sources of information, as well as any other sources mapped to the ontology, all of which can be shared with application partners more easily and with less duplication of effort. Consumer search and discovery applications can offer more consistent results with a common ontology, and more creative and engaging information from more data sources. For example, an ontology with commonly defined concepts for things like theme, subject, audience, and events, could be used to pull together keywords from more sources and organize them into data sets to drive more granular and more powerful consumer interaction with conversational discovery applications. Adoption of a public ontology even has the potential to empower an ecosystem where fans add back compatible data and information, which in turn leads to the creation of new applications and business opportunities.[3]

STRUCTURING AN ONTOLOGY FOR FILM AND TELEVISION – GENERAL CONSIDERATIONS

ONTOLOGY VS. SCHEMA – WHY CHOOSE AN ONTOLOGY?

To describe the technical underpinnings of an ontology, it is first necessary to ask, "Why an ontology rather than a schema or other ways of organizing data?" In theory an ontology and a schema (e.g., for XML or a relational database) both can describe the same thing. However, schemas tend to be designed for very targeted purposes (e.g., MovieLabs Common Metadata for consumer-facing metadata) and are good at well-defined problems with strictly structured data elements and requirements that are unlikely to change.

Ontologies are easier to extend for new concepts and are better at dealing with problems that are less well-structured. Extensions can be created as new parts of the ontology, or as external links to other ontologies, whereas schemas often inadvertently impose constraints that inhibit easy evolution. Structurally, an ontology sits above a schema, in that multiple schemas can be derived from the same ontology. Usually a schema derived from an ontology is specific to an application, and perhaps

[3] Another example is better understanding and use of genres. Currently, genre classification varies by studio, distributor, metadata provider, and ranking site. However, by using the ontology to collect and structure genres from multiple sources in multiple territories, an application can identify genre overlap and infer a work's perceived "basic" genre, providing more information for recommendations, bundled offers, and so on.

implements only part of the ontology to achieve a tactical purpose or target a specific application and set of technologies (e.g., SQL, graph database, document store).

Another advantage of an ontology is that it forces data technologists to confront the fundamental questions of "what is it?" and "what does it mean?" Schemas generally spend a great deal of time on the structure of the data and comparatively less on the innate nature of the data.

As a result, it is possible to create an ontology that covers an overwhelming number of concepts, classes, and relationships, all of which interact in precise ways, while maintaining fundamental extensibility. A schema of similar complexity would almost certainly be somewhat brittle, and tweaking one part will often produce unintended consequences for other parts, no matter how skilled and thorough the designer.[4] For similar reasons, it is much easier for an application to use a subset of an ontology, or a subset that has been mapped to a schema, than it is to use a subset of a schema.[5]

People provide a good example of how an ontology manages complexity. People have different names in different works, or even different names at different career stages. The name in a cast list may be a translated or transliterated version of a "real" name. The gender of a person or ways of talking about gender may change. Some information about people (country of birth, country of citizenship) may be useful for discovering affinities across movies, but not generally useful for presenting information to consumers. All of these complex and changing characteristics of people can be expressed cleanly in an ontology because the ontology treats people as separate entities,[6] allowing the internal complexity of people records to be hidden from the entities that are related to them. A Creative Work as an object no longer requires any details at all about a person who contributed to the work. Instead, it is sufficient to include information about how the person relates to the work (as a subject, as a contributor, and so on), which simplifies the model for the Creative Work significantly. It also allows implementations to use as much or as little of the full model's complexity as desired. Machine-oriented applications can use specific queries to extract simple information—"find all movies to which the person with ID Q24829[7] contributed as a director." Applications aimed at human users can extract additional information—"Orson Welles, b. 1915, d. 1985"—from the linked person record. Similarly, a query could ask: "find the people records for anyone named Alan Hale",[8] and if more than one record is returned,

[4] For example, the apparently simple exercise of adding a "romanized" attribute to EIDR title fields had unintended consequences in the names of people and organizations.

[5] See Appendix 2: The Real World for examples and more discussion of the practical applications of an ontology.

[6] "Peoples is peoples" – Pete, in *The Muppets Take Manhattan*.

[7] See https://www.wikidata.org/wiki/Q24829.

[8] Alan Hale Sr: Friar Tuck *in The Adventures of Robin Hood* (1938), Porthos in *The Man in the Iron Mask* (1939); Alan Hale Jr: The Skipper *in Gilligan's Island* (1964-1967), Porthos in *The Fifth Musketeer* (1979).

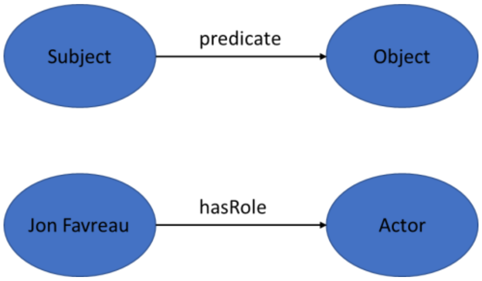
additional queries can target specific relevant people records or the specific works referenced in those records.

Ontologies and the data systems that use them also are good at using external identifiers, and there is now a sufficient network of resolvable work identifiers, especially from EIDR and Wikidata, to support building the concept in at a very basic level. In addition, other creative sectors have had good results using ontologies for purposes as varied as data aggregation from multiple inconsistent sources (one of our target use cases) and managing rights.[9]

Finally, consistent data sources with defined semantics and relationships are a significant resource for emerging systems based on machine learning and inference.

USING RDF TRIPLES WITH AN RDF ONTOLOGY

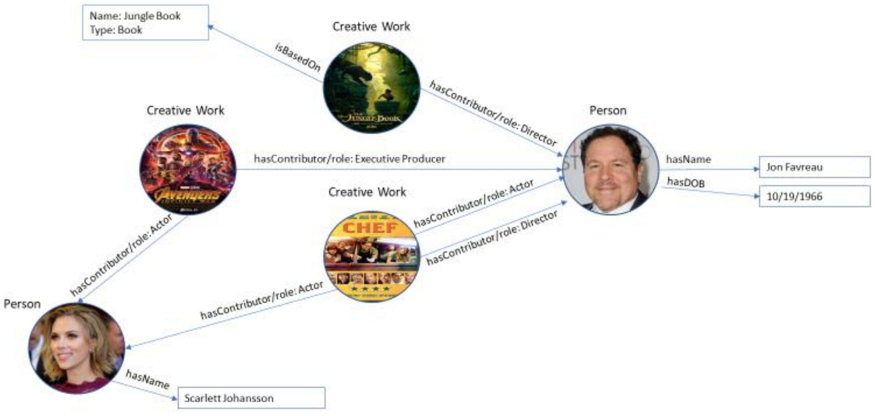
Both an ontology and commonly used expressions such as triples most often rely on the Resource Descriptor Framework (RDF) developed by the W3C. An RDF triple is an expression that contains a subject, predicate, and object—all expressed as RDF URI references. RDF triples serve as a useful way to include standardized relationships in databases of various kinds, especially graph databases. A simple example of an RDF triple would look something like this:



RDF triples also serve as the atomic elements of an RDF ontology. Triples can be used independently of an ontology, but an ontology provides a framework for connecting triples in a defined and structured way within and across databases. It standardizes the classes of objects and subjects described by triples, as well as the types of relationships that connect them. That standard framework then allows RDF triples to describe very specific relationships in a particular context, while maintaining conceptual consistency and interoperability with related groupings of triples in other databases. The ontology essentially ties together terms and vocabularies to avoid misunderstandings when data is

[9] For example, some Registration Authorities (RA's) for the ISO Digital Objection Identifier (DOI) standard are using linked data and webs of identifiers in their own sectors, especially print media.

aggregated from different sources. An example of using RDF triples to construct a few simple elements of an ontology might look like this:


Name: Jungle Book
Type: Book
Creative Work
isBasedOn
hasContributor/role: Director
Creative Work
Person
hasContributor/role: Executive Producer
hasName
Jon Favreau
hasDOB
10/19/1966
Creative Work
hasContributor/role: Actor
hasContributor/role: Director
hasContributor/role: Actor
Person
hasContributor/role: Actor
hasName
Scarlett Johansson

An ontology, of course, could be expressed conceptually in human-readable language without using RDF. A technical expression, however, has the advantage of enabling automation through machine-to-machine communication. Similar to the way XML enables greater automation than an Excel spreadsheet, even when both contain the same data, an RDF ontology enables more automated exchanges between different databases and sources.

CLEARLY DEFINED SCOPE OF COVERAGE

Ontologies are almost infinitely flexible, but that flexibility comes with a cost. An overly general ontology is hard to use as an interoperability tool, since implementations will tend to use their flexibility in whatever way suits them, and very general ontologies (which tend to be deeply nested) have problems with both intelligibility and performance. Conversely, an inflexible or overly restrictive ontology loses many of the intended benefits.

To reach a happy medium, an ontology needs a clearly defined scope and level of granularity within the given domain. Defining the scope means that some things can be ignored, which is always good (remember, you can always add them later). Any concepts with uncertain scope or granularity (franchise vs brand, for example) can be turned into problems of terminology and definition, rather than structural problems, and it's relatively simple to add terms and definitions.[10]

Our scope was limited primarily to movies and TV shows. We focused more effort on movies, which also addresses much of TV, and less on work that would be specific to TV. As anyone who has built a system that handles both will tell you, TV is harder, so we have only a skeletal mechanism for episodic TV, which can be expanded as needed.

In other areas, we attempted to enumerate exactly what would be included and what would not. Things that we explicitly included are:

- The sources of the data, which we deemed important since data aggregation is a key initial use case. Additionally, pulling in multiple sources of the same kind of data can be particularly helpful, especially when the sources vary across geographies.
- Modelling of factual information about a work (e.g., cast, crew, release dates, awards, ratings). Many schemas cover this information, but we have not yet found an ontology that manages it with a useful level of precision and flexibility.
- Identifying the work via alternate titles and identifiers, which is useful input for data merging applications and improves connectivity to other data sources.
- People as they relate to a work. We did not try to model everything about a person, but only things that are of significant interest relative to a work.
- Data that differs from country to country (or by language), which is a current pain point according to input from industry stakeholders. Country-specific data is often found in separate, incompatible databases.
- External numeric data (e.g., rankings, aggregated consumption data).
- Common textual data covering things like genre, subject, and theme.
- Links to other kinds of data, such as reviews, synopsis, and artwork.
- Generic grouping, with ways to create specific types of groups to cover brands, franchises, themed collections, and so on. (See below under Groups for more details.)
- Relationships of a work to other things; we narrowed this to a work's relation to other works (sequels, prequels, parodies, reboots, and so on) and to things a work might be based on (e.g., books, plays, games)

[10] However, it is important to define terms clearly and precisely. Even apparently simple things like "role" or "rating" must be defined carefully, since they can mean different things to different people. Definitions can be expanded or restricted based on particular uses of the ontology and model.

The first release focused to a lesser degree on the areas below, although work in each area can be expanded as future needs dictate:

- Deciding whether things are canonical or authoritative. The model can identify the source of data, so an application or user can decide which sources to trust, how to rank them, and so on. An application also can take in data from multiple sources, analyse it, and add the result as a new piece of data with a new source indicating the data has been massaged in some way, e.g., "MovielabsInferredGenre".
- Edits and versions of a work. This is an obvious extension to consider, which can then be connected to ontologies that deal with frames and scenes, which are dependent on a particular version or manifestation of a work.[11]
- "Complete" person information. (See above.)
- Character data. The model currently describes a portrayed character as a simple text string. It would be easy to extend the model with a full character-based ontology (e.g., Young Indiana Jones, Pre-WWII Indiana Jones, Cold War Indiana Jones) or connect to a pre-existing character ontology.
- Individual consumer behaviour. This type of data can be added later.[12]

THE ONTOLOGY

THINGS WE CARE ABOUT – ENTITIES, OBJECTS, TOP-LEVEL CLASSES

This white paper describes an ontology for the film and television industry. The "entities" we care about, therefore, are creative works and the concepts necessary to describe creative works and their connections. Those concepts serve to organize the objects (or "things") that one naturally refers to when talking about creative works. Objects are organized into classes (and subclasses) and have properties. Objects also have relationships with other objects that are represented in the ontology as connections (or predicates). The sections below describe the most important top-level classes of objects and common concepts that frame our creative works ontology.[13]

[11] For example, in addition to contributing valuable parts of this ontology, Disney/ABC has created a very useful ontology for frame and scene-based metadata.

[12] An ontology can model different varieties of data on consumer behaviour reasonably well, but most systems that consume this data prefer tabular form, so there is some research required to find a good middle ground.

[13] More detailed information on all the classes and subclasses that make up the ontology can be found in the technical documentation published with the ontology itself.

TOP-LEVEL CLASSES

This section covers classes that we expect to be first-class objects, i.e., they can exist on their own, without being attached to other objects.

CREATIVE WORK

In addition to areas of focus, one of the most important design decisions with any ontology is where and how to accommodate future extensions and other uses. We focused initially on both film and television, but quickly decided to concentrate on film. However, since at the most fundamental level film and television share many characteristics, we chose a more general term—Creative Work—for the top-level class of film and television works.

"Creative Work" has the advantage of being true, descriptive, and, at the level of the ontology, unambiguous. Some systems call a movie a "title", but title is much more useful as the name of "what the movie is called", and it is rarely used for television works. "Creative Work" is also all-encompassing, with related properties or fields that apply to other kinds of Creative Works as well: contributors (authors for books, songwriters and performers for music), associated locations (place of publication of music, setting or place of publication for books), and so on.[14]

To move from the more general "Creative Works" to specific types of works, the ontology uses Scopes, a mechanism developed by Disney/ABC, to differentiate between film, television episodes, television series, and versions. The specific details for each type of work are pushed into a Scope instance connected to the Creative Work. For this version of the ontology, we concentrated on fleshing out the Scope instance for films, with only skeletal Scopes for episodic television. Except for the separate specifics of each Scope instance, however, everything in this ontology is directly applicable to both film and television, or can use subclasses and types to allow for differences. For example, both have Contributors (actors, directors, producers, and so on), although film and television also have separate contributor types as well. Similarly, both have Awards, but Oscars apply to film and Emmys apply to television, so the Award class is intended to be sub-classed.

Creative Works have the following properties or fields:

- Identifier: Identifiers serve two purposes. They provide more reliable identification than simple matching based on other fields (as long as the identifier source is trustworthy, so the identifier

[14] There is precedent for this level of generality. The Dublin Core ontology was designed to cover almost all creative things, but is so high-level that it cannot be used well for many concrete applications, and there are many cases of competing industry extensions for it. As another example, MovieLabs Common Metadata can cover non-audiovisual works, but is not generally used for non-audiovisual types.

element includes source attribution). In addition, some of them – resolvable identifiers – provide links to other systems. (See the "Identifier" section below for more details.)

- Award: Awards are also discussed more fully below.
- Keyword: This is a generic class that contains a type (genre or subject, for example), a value ("Comedy", "Space exploration"), and a weight (primary and secondary genre, or 80% romantic and 30% dramatic). Keywords are attributable, and so have a source.
- Rating: This element is for censorship and audience suitability ratings, not "rated 4 stars", which is covered under "Ranking". Ratings are attributable and are an RDF version of the MovieLabs Common Ratings specification.
- Title: Creative Works have many titles, and many kinds of titles. All titles are attributable and can indicate a language and a country. There are a few subclasses of Title, including "Release", "Translated", and "AKA".
- Ranking: A ranking is a weighted value from a source, e.g., "80% on Rotten Tomatoes" or "4.5 stars on IMDB". The ranking itself has a source ("This rank came from IMDB"), as does its connection to the Creative Work ("This IMDB ranking came from AlloCine").
- originalLanguage: The original language of the Creative Work. This field has a source, since not all sources agree on the original language of a work.
- originalReleaseYear, originalReleaseChannel: These can be derived from Release elements (see Release discussion below), but it is convenient for many applications to have them readily accessible. The source will usually be something like "internal" or "computed", rather than taken from one specific source.
- associatedLocation: This field links to the subclasses Narrative Location, Filming Location, and Production Country. (See below for a general description of a Location.)
- approximateLength: This is an attributable duration, normalized to the RDF/XML "duration" type.
- Cost: Attributable. Described with an amount and a currency.
- relatedCompany: This is an attributable reference to a Company. A Company has identifiers, a name, and a type, e.g. "producer" or "distributor." "Production Company" and "Distributor" are derived classes.
- basedOn and relatedTo: Creative Works often are connected to other Creative Works and lots of other things. (See below for how this is managed.)
- Text: This is an attributable field that contains longer free-form text, plus a country, a language, a date, and a type. We have defined derived types for Synopsis and Review.
- Contributor: Anyone who has participated in the making of the Creative Work. Many systems separate contributors arbitrarily into cast and crew, but there is enough commonality to have a single structure with internal differentiation. If a person did more than one thing, he or she will have two Contributor records, each of which links to the same person. For example, an actor/director should have one record for each job, and an actor who portrays more than one character should have one record for each character. The Contributor field itself is not

attributed, though individual elements within it are. Elements within the Contributor field include Job, Type (cast or crew), Person, Billing and Portrayal. (See appendix for additional detail.)

- Release: A Release has information about when and how a Creative Work has been released and can have information about consumption of that release. It includes:
    - The start and end dates of the release.
    - The country of countries in which the release occurred.
    - Distributor
    - Channel: Includes Theatrical, Broadcast, Home, Piracy, etc.
    - Distribution Model: Gives details of how the work was distributed in the Channel, and includes, for example, Disc, SVOC, PVOD for Home, BitTorrent for piracy, Premier, Wide, and Festival for Theatrical.
    - Format: Covers things like DVD, Blu-Ray, 3D, 70mm, and iMax.
    - Consumption: The ontology covers only aggregated consumption data, e.g., ticket sales or box office receipts, and does not cover individual consumer data. Consumption includes an amount and the type of unit in which that amount is expressed (e.g., tickets, downloads, dollars.) A Consumption record can also include a date, date range, or date and duration, since the time period for which the consumption is measured may not be the entire duration of the Release. A Consumption record also can have sub-records. For example, a Release can have a consumption for total box office, which in turn has sub-records for each of the first three months.

PERSON

A Person represents a person independent of any role in a work. Multiple Creative Works can refer to the same Person; for example, an actor in one film might be a director in another. The ontology represents only the aspects of a person that are needed when thinking about a Creative Work, based on a survey of various use cases.

Generating Person records can be very complex because only a few systems (e.g., IMDB, Wikidata, BFI, and ISNI) have proper databases of people, and most of them – with the partial exception of ISNI – do not connect easily to other systems. Indeed, most systems that describe Creative Works provide only a name for a Person and possibly an internal identifier, such as an IMDB nm code or a Wikidata ID. Combining person data from multiple sources using names is somewhat simpler than combining Creative Works using titles—because the matching can be done in the context of a work, rather than in the whole universe of person names – but it is still quite difficult to implement consistently.

The Person class has the following properties or fields:

- identifiers: These are any external identifiers available for this person.

- preferredName: The (relatively simple) name by which the person is generally known in a preferred language.
- names: This uses the PersonName class to represent other names for the person. It includes a source, a language, and a type. PersonName is not sub-classed currently, but there are instances of PersonNameType for AKA, Billed name, Contractual Name, Credited Name, Real Name, and Translated Name. Some or all of these may become subclasses of PersonName (based on industry agreement), with type being used for other less common cases. Management of multiple names for an individual can be complex, and we fully expect this simple solution to be refined as needed, e.g., by attaching a billed name to a particular Contributor record in a Creative Work.
- countryOfCitizenhip, birthplace: These are both Locations (see below).
- seenAsStar: Refers to an attributable class, indicating that the person is ranked as a star by that source.
- gender: Gender is more complicated than simple male/female and is divided into two properties: hasGender, which is the person's self-identified gender, and isTransgender, which is a Boolean value. This model is compatible with the model in Common Metadata 2.8 and was defined after consultation with interested parties in the US and the UK.
- dateOfBirth, dateOfDeath: Both are attributable dates.

LOCATION

A Location represents a real or fictional location. It is used in many places in the ontology (filming location, production country, setting, and birthplace) and can have the following fields:

- identifier: The identifier in a particular system for this Location. An implementation may decide not to implement locations as separate objects, especially if it is not in RDF; in that case, location information will be in-line with records that refer to it, resulting in some duplication but perhaps a simpler database structure.
- name: The name by which this Location is commonly.
- country: The name of the country of the Location, if applicable.
- countryCode: A country code for the Location, if applicable. Strictly speaking, if countryCode is present, Country is unnecessary since there are public systems for looking up country codes, but many systems will want to include both for performance reasons.
- locationDetails: Any further information about the location.
- coords: A latitude/longitude pair.
- landmark: An attributable field indicating that the source has marked this location as a landmark (e.g., The Eiffel Tower or the Trevi Fountain).

- fictional: A Boolean property; if true, the location is fictional. For example, this would be set to true for Freedonia, the Duchy of Grand Fenwick, and Tatooine. In general, fictional locations will be used only as settings.[15]

GROUP

A Group is just a collection of Creative Works. Some Groups can be inferred by using other queryable information, such as genre or character, but extended inferences can cause performance problems, particularly in strict triple-based implementations, and missing data may cause incomplete results. Furthermore, it is not always possible to infer membership in a Group from other metadata (unless you add new keywords and keyword types, which rapidly becomes unmanageable and non-intuitive.)

There appears to be a semi-consensus on some types of Groups, especially Brand, Universe, and Franchise, but much less agreement on what those types precisely mean. Since this is an ontology, new types can be added based on industry consensus. Strict definitions of current and proposed types can also be added as developed.

The types in the ontology now are:[16]

- Universe, Brand, and Franchise, the definitions of which are still fuzzy. It's clear that Star Wars, Star Trek, Marvel Cinematic Universe, The Godfather, Lego, and Bond (James Bond) somehow fit in those loosely defined categories, and we expect more precise definitions to be added based on industry agreement.
- Character, which is a Group containing films that share a character (e.g., a Sherlock Holmes Group that contains The Hound of The Baskervilles (1939), The Seven Percent Solution, Sherlock Holmes (2009), and The Private Life of Sherlock Holmes. A Batman Character group can contain all the usual Batman movies and TV shows, as well as Lego Batman.
- Ad Hoc, which are groups created by some other means, possibly thematic or based on intended audience. Teen Dystopias, Oddball Superheroes, and Pirates are good examples of candidates.

Because Groups are implemented as a class, rather than as a collection of tags and properties, they are easier to manage and have an existence of their own. Groups contain only Creative Works, so they're not nestable and there is no explicit or implied hierarchy. This is because all the movies in the Batman Character Group may not be in the DC Brand or DC Cinematic Universe, for example.

[15] However, if the Character model is expanded, a fictional character may have a fictional birthplace – Superman was born on Krypton – or may not: Indiana Jones is reported to have been born in Princeton NJ, and James T Kirk was born in Iowa.

[16] Groups for 'Sales' and 'Display' collections are included as placeholder types, but not implemented except as comments in the Ontology.

Groups have straightforward properties, even if the concept is sometimes confusing because of preconceived notions:

- Type: The type of the Group, as described above.
- isOfficial: A Boolean indicting whether the Group has some form of official or canonical nature. Applications then can decide whether to use the Group based on criteria of the use case (e.g., analytics applications may have different criteria than consumer-facing applications.
- Source: Groups are Attributable, and Source identifies the origin of the Group, i.e., who created the grouping in the database.
- Identifiers: External identifiers for the Group, using the standard Identifier structure. Public Identifiers for Groups are almost nonexistent, but some studios have internal identifiers.
- Description: A description of the Group.
- Note: Other information not in the Description.
- Members: Each member is the id of a Creative Work; membership in the Group is attributable, and if the source is present, it indicates who assigned the work to the Group.

AWARD

Awards are significantly more complicated than one might think, and the framework for awards in the current ontology is continuing to evolve. At a simple level, however, the ontology contains an Award class with basic information such as:

- Year: The year the award was presented.
- Sequence Number: Many awards are referred to as "The 60th Annual…" or similar, and this information is hard to derive otherwise.
- Details: Further details about the Award, e.g., "75th Academy Award for Best Director".
- Type of award: In the current version, this is inferred from the entity to which the award is connected, e.g., "Best Picture" for an award connected directly to a Creative Work. The next step is to model subclasses that can collect similar types of awards. For example, the Best Picture Academy Award has gone under a variety of names, and Best Documentary and Best Foreign Film (for the Academy Awards) can be viewed as subclasses of a "Best Film (any kind)" category.

There also are subclasses defined for some common awards, e.g., Oscar, Emmy, BAFTA.

Awards can be attached to a Creative Work, a Contributor, a Company, or a Person through the use of two relationships:

- Is Nominated
- Is Winner

RELATED THING

One of the most basic tasks of an ontology is to connect things to other things. Some kinds of data are innately connected to a Creative Work, e.g., Contributors, Locations, and Awards. Beyond innate connections, we limited the scope to two concepts—RelatedTo and BasedOn.

- RelatedTo covers relationships to other Creative Works and is intended to be sub-classed/extended to cover common concepts such as prequel, sequel, reboot, parody, and the like.
- BasedOn covers a Creative Work's derivation from other things, such as books, comics, theme park rides, characters, and games.

RelatedThing is an umbrella class for the similar, but different, concepts of RelatedWork and BasedOnThis, which are respectively the classes used as the objects of relatedTo and basedOn.

- The RelatedWork class defines a Creative Work to which some other Creative Work stands in some relation and is the object of the RelatedTo predicate.
- The BasedOnThis class defines an entity that served as the basis for a Creative Work, e.g., a novel that is the basis for a Creative Work, and is the object of the BasedOn predicate. Even though such entities are creative works in a more general sense, within this ontology the notion of a Creative Work is limited to a movie, video, TV program, or similar audiovisual work.

The ontology supports both of these predicates and classes, and it is our intent to extend it to deal with terms and meanings for RelatedTo, and a usefully large but tractably small set of BasedOnThis subtypes.

COMMON CONCEPTS

IDENTIFIER

Ontologies place special emphasis on connections. This ontology uses identifiers to connect data to external sources that describe the same thing. Although there are issues – even good identifier systems have mistakes, duplicates, and deprecated records – this approach is nonetheless more reliable than basing external connections on string matching (e.g., for names or titles.) An implementation of the ontology may use a combination of identifiers and string matching to merge data from multiple sources.

Not all classes of objects have reliable identifiers. Creative Works have many to choose from: EIDR, IMDB, ISAN, Wikidata, BFI, TMDB, etc. EIDR and Wikidata are especially useful because they provide alternate identifiers for the work. People are less well served; IMDB, Wikidata, and BFI are a good starting point for person identifiers, but they can be unreliable and vary in how they deal with real names, "billed as" names, and the like; links to other systems are generally less good. Despite these issues, it is important to take advantage of identifier-based connectivity whenever possible.

The ontology defines subclasses for common Identifier types.

ATTRIBUTABLE

Because an ontology is often used for collecting data from multiple sources, it is essential to keep track of provenance – where the data came from. The ontology defines an Attributable class for this purpose.[17]

Implementations of the ontology can also use attribution to indicate when values are synthesized or inferred. For example, a system that gathers release dates from multiple sources can attribute an earliest known release date to a named source, such as "inferred", indicating that the value is computed from the raw data. The naming of synthesizing sources is outside the scope of the ontology itself.

COUNTRY AND LANGUAGE

Many pieces of data are expressed in a particular language; titles and names are the most common examples. Some data has an explicit relationship to a country, such as box office data or consumer reviews. Some items, such as a synopsis, can have both a country and a language. The ontology includes a language attribute for every text-based element, and a country attribute where appropriate.

Implementations can refine these attributes. Easy extensions would include adding a special country code for worldwide data (such as global box office) and refining the "Anything with a country" class to include regions as well as countries.

We recommend that implementations follow EIDR practices for languages (https://eidr.org/assets/Using-EIDR-Language-Codes-v1.9.pdf) and countries/regions (https://eidr.org/assets/Using-EIDR-Region-Codes-v1.1.pdf).

SCOPE

The "Scopes" concept used in this ontology is based on a closely related ontology created by Disney-ABC.

Using Scopes, a Creative Work can be broader or narrower than another Creative Work, or it can be linked indirectly by contributing to a higher-level grouping. Instead of listing out specific subclasses of

[17] Any instance of a class whose elements are expected to come from the same source inherits from the AttributableClass; i.e., if the elements of an object are considered an inseparable bundle, the object is attributed, not the elements. Any element of an object with multiple sources (e.g., titles, names of people) is individually attributable.

Creative Work, the ontology instead assigns a Creative Work a "Scope", whose properties then define specific types of Creative Works. In the code this would look like:

```
CreativeWork123 a CreativeWork .
CreativeWork123 hasScope Scope123 .
Scope123 hasProperty  Property1, Property2, Property3 .
```

A simple Scope can be defined for a Movie, a TV Movie, a Direct-to-Video Movie, and an Episode without any impact on the underlying definition of a Creative Work. Scopes can also be used for edits and versions of a Creative Work.

Separating Scope from Creative Work allows greater flexibility to define the specific characteristics of each type of Creative Work without affecting the broader/narrower semantics of the Creative Works themselves. The advantages of this construction are:

(a) preserving the hierarchy and relationships between works without having to add complicated restrictions in the base model;
(b) avoiding potential conflicts between properties that are not (and should not be) shared between all types of Creative Works;
(c) defining Scopes on an as-needed basis without worrying about any effects on the underlying Creative Works.

MISCELLANEOUS CONVENTIONS

In general, the ontology uses subclasses to refine concepts. For example, the Title class has subclasses for original and other official titles. Classes that are subclassed also have a "type" attribute to allow for refinements that do not have explicit subclasses, which can occur in two cases: when a type has subtypes for which there is no general agreement on standardization; and when subtypes come in from systems that we have not explicitly considered. Using Title as an example, EIDR has an extensive list of title types, some of which overlap with title types from, for example, IMDB, without being exact semantic matches. An implementation can decide to add its own subclasses, or just use the type field for these extensions. This kind of extensibility is significantly easier in an RDF ontology than in a traditional XSD schema.

The ontology does not enforce required vs optional fields; an implementation may decide to do so, or add other constraints as well, e.g., disallowing country codes for fictional locations.

CONCLUSION

Data is certainly one important key to the future of the film and television industry. This white paper describes one significant way to improve the core data infrastructure of the industry—a shared industry creative works ontology. A common ontology has the potential to support and advance multiple components of the industry's core infrastructure—enterprise data systems, analytics and marketing systems, data warehousing applications, and almost any other data system that relies on integration of data from the many independent sources around the media industry. It does so by delivering a machine-to-machine framework for automating connections between systems in a way that enables greater interoperability, faster and more efficient application development, and new capabilities that otherwise could be too burdensome to implement. It is core data plumbing that will benefit the entire industry. We look forward to working with industry stakeholders to further understand the potential applications of a common creative works ontology in order to help build a better and more competitive data future.

APPENDIX 1: RESOURCES

MovieLabs has also built a prototype implementation of the ontology to confirm the technical correctness of the ontology and test potential applications. The prototype will be made available to the industry for testing and further development of the ontology and applications. MovieLabs also plans to release a number of mappings between the ontology and other commonly used systems to aid in the development of applications.

This white paper and the full creative works ontology with documentation is available at www.movielabs.com.

APPENDIX 2 - THE REAL WORLD: NOTES ON ADOPTION AND PRACTICALITY

There is little point in doing a new ontology if it is not used by its target audience. The uptake of new infrastructure specifications can seem glacially slow. For example, EIDR, MDDF, and Common Metadata (all Movielabs specifications) have taken several years to get into the mainstream.

Ontologies have historically had very slow adoption for multiple reasons, some based on mindset and some based on technology. All ontologies need a strong philosophical and theoretical underpinning, but early in the development of RDF ontologies the theoretical aspects were too prominent, acting as a barrier to understanding and adoption – the underpinnings obscured the purpose. As ontologies (and ontologists) have matured in both theory and practice, the philosophical underpinnings have become less of an obstacle. Gradually, more subject matter experts (the people an ontology is supposed to help) have become ontology-literate, and ontologists have become more aware of the real world.

A second cause of slow adoption has been slow performance of the underlying implementation technology. The first choice for implementing an ontology tends to be a triple store; the performance of triple stores has improved dramatically, but for several reasons may never equal the performance of a relational database that has been optimized for a particular problem: generality is computationally expensive. However, a system that implements an ontology does not have to be a triple store. The prototype built by MovieLabs was initially a pure triple store, but migrated to a mixed document store/triple store model for performance reasons. The current prototype has sacrificed some small sliver of querying capability for obscure cases, but the performance is more or less equivalent to that of a more traditional database, all while retaining the flexibility of the ontology's connections and looser structure. The mixed model prototype also produces XML and JSON output quite easily, an advantage for application developers who do not commonly rely on RDF.

Finally, triple stores use the SPARQL query language, which is incredibly powerful and incredibly opaque, with a very steep learning curve (although many of its advanced features mimic analogous features of SQL). SPARQL is a joy to use if you become at one with it, but that's asking a lot. However, newer query languages, such as GraphQL (which the MovieLabs prototype uses), are somewhat easier to use and can provide the interesting features of SPARQL queries—such as easy following of nested relationships, simple filtering based on multiple criteria and multiple fields, easy handling of grouped and nested results, and coping with missing data much more simply. The result is a happy medium between the painful power of SPARQL and the straightjacket of SQL.

In the end, utility trumps inertia, but utility is in the eye of the beholder. "Useful" means different things to different people: for some, it is entirely a cost/benefit calculation; for others, it means enabling new classes of applications; for still others, it means a radical improvement in automation or communication. The list of possible applications covers all these potential utilities and more – precisely because an ontology has the advantage of being more adaptable than other data definition techniques.

DDEX, which the music industry uses extensively for automation of sales, reporting, and rights clearance workflows, is based on an underlying ontology. The ontology is never used as-is, but "slices" are taken and converted to application-specific XSD/XML. Because all the schemas are derived from the same ontology, interoperability is very good across all the workflows. The ontology is very flexible, but with enough formal rigor to target multiple applications and implementations without reinventing the wheel each time. In addition, using only the necessary parts for each XML schema simplifies the schemas and speeds adoption.

The Linked Content Coalition (LCC) provides an ontology for use in rights management. It has minimal descriptive information for the works being licensed, relying instead on reliable, resolvable identifiers (such as EIDR and DOI.) It is formally defined as RDF. One implementation (the UK Copyright Hub, which facilitates licensing of high volume, low value content) uses triples directly, but other implementations, such as the European ARDITO project and mEDRA's licensing system for Italian publishers, use an XML version. Mappings have been done to different rights expression languages, including ODRL from W3C.

Both DDEX and LCC have succeeded because the ontologies were designed by a mix of ontologists, experts in the various fields, and implementers. This is the example that we have tried to follow with the Creative Works ontology that is the subject of this white paper.

APPENDIX 3

For readers interested in greater detail, additional information on some classes or elements of the ontology is included in this appendix. For complete detail, see the published ontology and its documentation.

CONTRIBUTOR FIELD – ADDITIONAL DETAILED ELEMENTS

- Job: Actor, Director, Producer, Screenwriter, etc. We do not enumerate jobs exhaustively, but have subclasses for the more common jobs, and future work will introduce the OWL and SKOS mechanisms to map to other job code systems, such as EBU Core.
- Type: Cast or crew. This can be derived easily (anyone who is not an Actor is crew, and Actors are cast), but a separate element is included because some implementations may want to cache this information.
- Person: The Person record for this Contributor. (See Person discussion below.)
- Billing: The name of the person as billed in the movie, plus a source. Can also contain a BillingOrder, most commonly used for actors, but sometimes used for movies with multiple producers or directors as well.
- Portrayal: Applies only to Actor (and of course any classes derived from it, such as Voice Actor.) A portrayal lists a source, the name of the actor, as billed, and a reference to the character portrayed. We have defined a Character class that currently has only a name attribute. We anticipate linking to other more developed character ontologies or defining a fuller Character class with a mapping to other ontologies.