

IEEE Ontological Standard for Ethically Driven Robotics and Automation Systems

IEEE Robotics and Automation Society

Developed by the
Standing Committee for Standards Activities

IEEE Std 7007™-2021

STANDARDS

IEEE Ontological Standard for Ethically Driven Robotics and Automation Systems

Developed by the

Standing Committee for Standards Activities
of the
IEEE Robotics and Automation Society

Approved 23 September 2021

IEEE SA Standards Board

Abstract: A set of ontologies with different abstraction levels that contain concepts, definitions, axioms, and use cases that assist in the development of ethically driven methodologies for the design of robots and automation systems is established by this standard. It focuses on the robotics and automation domain without considering any particular applications and can be used in multiple ways, for instance, during the development of robotics and automation systems as a guideline or as a reference “taxonomy” to enable clear and precise communication among members from different communities that include robotics and automation, ethics, and correlated areas. Users of this standard need to have a minimal knowledge of formal logics to understand the axiomatization expressed in Common Logic Interchange Format.

Keywords: artificial intelligence, automation, ethics, IEEE 7007™, ontology, robotics

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2021 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 12 November 2021. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-8013-0 STD24973
Print: ISBN 978-1-5044-8014-7 STDPD24973

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE does not warrant or represent the accuracy or completeness of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to results and workmanlike effort. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and Standards Coordinating Committees are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the [IEEE SA myProject system](#). An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These

include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#). For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#). Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting

inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this IEEE standard was completed, the Ontologies for Ethically Driven Robotics and Automation Working Group had the following membership:

Edson Prestes, *Chair*
Sandro Rama Fiorini, *Vice Chair*
Mike Houghtaling, *Technical Editor*
Babita Ramlal, *Technical Editor*
Paulo Jorge Sequeira Gonçalves, *Secretary*

Bernd Blobel
Don Brutzman
Joel Carbonera
Abdelghani Chibani
Patrick Courtney
Nicola Fabiano
Tamas Haidegger

Vicky Hailey
Dennis Holstein
Tom Kurihara
Zvikomborero Murahwi
Joanna Isabelle Olszewska
Brian Page

William Remington Patterson
S. Veera Ragavan
Randy Rannow
Martin Saerbeck
André de O. Schenini Moreira
Ozlem Ulgen
Altaz Valani

The following members of the individual Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Robert Aiello
M. Victoria Alonso
Pieter Botman
Diego Chiozzi
Murphy Choy
Jennifer Costley
Nicola Fabiano
Luis Andres Fajardo Arturo
Sandro Rama Fiorini
Paulo Goncalves
Didem Gurdur Broo
Tamas Haidegger
Robert Hobbs
Werner Hoelzl
Michael Houghtaling
Faiz Ikramulla

Piotr Karocki
Edmund Kienast
Robert Kozma
Loi Lei Lai
Sean Laroque-Doherty
Ting Li
Xiao Liang
Lars Luenenburger
Javier Luiso
Quintin McGrath
Lingzhong Meng
Lyria Bennett Moses
Rajesh Murthy
Alexander Novotny
Joanna Isabelle Olszewska
Brian Page

Sridhar Raghavan
Randy Rannow
Annette Reilly
John Sheppard
Matthew Silveira
Barnaby Simkin
Wayne Stec
Robert Soper
David Tepen
Ozlem Ulgen
Ionel Marius Vladan
Eleanor Watson
Stephen Webb
Robert Wortham
Naritoshi Yoshinaga
Yu Yuan

When the IEEE SA Standards Board approved this standard on 23 September 2021, it had the following membership:

Gary Hoffman, *Chair*
Jon Walter Rosdahl, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Edward A. Addy
Doug Edwards
Ramy Ahmed Fathy
J. Travis Griffith
Thomas Koshy
Joseph L. Koepfinger*
David J. Law

Howard Li
Daozhuang Lin
Kevin Lu
Daleep C. Mohla
Chenhui Niu
Damir Novosel
Annette Reilly
Dorothy Stanley

Mehmet Ulema
Lei Wang
F. Keith Waters
Karl Weber
Sha Wei
Howard Wolfman
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 7007-2021, IEEE Ontological Standard for Ethically Driven Robotics and Automation Systems.

Ontologies are formal specifications of a shared conceptualization denoting relevant concepts and relationships for a target domain of discourse. The set of concepts and relationships selected for formalization are those deemed appropriate for the type of ontology and its intended use. As a type of model, ontologies are abstractions of reality for the domain of interest and are comprised of classes, attributes, relationships, constraints, rules, and axioms. These components are used to express the shared commitments specified by the vocabulary of terms defined in the ontology. To enable viable ontology reuse, ontological models frequently utilize an architectural framework of three model levels: a top or foundational level, a middle (core) level, and a domain-specific application level. The ontology model for this standard is positioned as a core level specification. For more about ontology architectures, levels, and types see Aßman, Zschaler, and Wagner [B9] and Guizzardi [B25].¹ The use of ontologies for representing knowledge in any domain enables the following:

- Clear and formal definition of concepts for a domain
- Analysis of concepts and their relationships in searching of inconsistency, incompleteness, and redundancy
- Establishment of a language for use in the communication process among robotic and non-robotic systems from different manufacturers and among different stakeholders

The increasing complexity of robot design, human-robot interaction, and the increasing proliferation of robots in societies requires ontological standards to move beyond concepts and create unambiguous taxonomies with properties and examples of practical applications in defined use cases.

In 2015, the IEEE Robotics and Automation Society published its first standard: IEEE Std 1872™-2015, IEEE Standard Ontologies for Robotics and Automation. This standard established a series of ontologies about robotics and automation (R&A) to represent knowledge in R&A domain through a common set of terms and definitions that allows for explicit knowledge transfer among any group of humans, robots, and other artificial systems. Among these ontologies, Core Ontology for Robotics and Automation (CORA) was developed to be a high-level standard from which domain-specific efforts could emanate. CORA has generic concepts of the R&A domain (e.g., robot, robot group, and robotic system) to serve as basis for more focused ontologies in R&A to address specific information requirements.

IEEE Std 7007 complements IEEE Std 1872-2015 by focusing on the R&A domain, taking in into consideration the ethical dimension without being constrained to any application or kind of robot. In addition, IEEE Std 7007 complements the IEEE 7000 series of standards, such as the IEEE Std 7000™, IEEE Standard Model Process for Addressing Ethical Concerns during System Design, which focuses on ethical considerations at each phase of development to prevent negative or unintended consequences.

The creation of IEEE Std 7007 is timely, as the need for the ethical creation and use of R&A technologies has emerged as a vital aspect of advancing the well-being of human beings. Designers and manufacturers are increasingly giving robots the capability to interact and collaborate with humans, preferably in an ethical and safe manner to meet the needs of modern industry and societal advancements. As the complexity in design and new applications of R&A Systems arise, it is important for standards bodies to keep pace with advancements in the field in order to guide robot activities, support progress, and evaluate these for conformity and acceptability by society.

¹ The numbers in brackets correspond to those of the bibliography in Annex E.

In order to aid in the creation of value-sensitive design in R&A and align with what stakeholders (government, industry, academia, civil society) expect, in terms of benefits and a positive impact on human well-being, the consideration of applied ethics and a multidisciplinary approach to the standard development has been undertaken, with input from legal, philosophical, and engineering experts for this work. Thus, it is expected that IEEE Std 7007 can be used in multiple ways, for instance, during the development of R&A systems as a guideline or as a reference document to enable a clear and precise communication among members from different communities that include robotics and automation, ethics, and correlated areas of expertise.

Contents

1. Overview	11
1.1 Scope	11
1.2 Purpose	11
1.3 Word usage	12
2. Normative references.....	12
3. Definitions, acronyms, and abbreviations	12
3.1 Definitions	12
3.2 Acronyms and abbreviations	13
4. Ontologies for ethically aligned robotics and automation systems.....	13
4.1 Conventions	13
4.2 Background.....	14
4.3 Top-level definitions.....	15
4.4 ERAS top-level concepts.....	16
4.5 Norms and Ethical Principles	25
4.6 Data Privacy and Protection	40
4.7 Transparency and Accountability	57
4.8 Ethical Violation Management	69
Annex A (informative) Informative definitions.....	82
A.1 Top-level definitions.....	82
A.2 Norms and Ethical Principles.....	83
A.3 Data Protection and Privacy	88
A.4 Transparency and accountability	95
A.5 Ethical Violation Management	100
Annex B (informative) Ontology development.....	102
Annex C (informative) Use cases	104
C.1 Use Case Template	104
C.2 Norms and Ethics Use Case: Domestic Personal Assistant Robot.....	105
C.3 Ethical Violation Management Use Case: Data Privacy and Protection.....	107
C.4 Transparency use case: autonomous system behavior explanation.....	109
Annex D (informative) Distributed Responsibility Ascription for Autonomous Systems	112
Annex E (informative) Bibliography.....	114

IEEE Ontological Standard for Ethically Driven Robotics and Automation Systems

1. Overview

1.1 Scope

This standard establishes a set of ontologies with different abstraction levels that contain concepts, definitions, axioms, and use cases that are deemed relevant and appropriate to establish ethically driven methodologies for the design of robots and automation (R&A) systems.

1.2 Purpose

The purpose of the standard is to establish a set of definitions and their relationships to enable the development of R&A in accordance with shared values and internationally accepted ethical principles that facilitate trust in the creation and use of R&A. Emphasis is placed on the alignment of ethics and engineering to enable communities to understand how to pragmatically design and implement these systems within the context of a values-based society. These definitions allow for precise communications among global experts of different domains that include robotics, automation, artificial intelligence (AI), and ethics.

The use of ontologies for representing knowledge in any domain has several benefits that include the following:

- a) A formal definition of concepts of a particular domain in a language-independent representation, i.e., they are not dependent on a specific programming language, however, they are formally described to be implemented in a target language
- b) Tools for analyzing concepts and their relationships in searching for inconsistency, incompleteness, and redundancy
- c) Language being used in the communication process among robots from different manufacturers

Users of this standard are responsible for being apprised of and referring to appropriate, applicable ethical criteria for consideration during system design. Moreover, users should consult all applicable laws and regulations. Conformance to the provisions of this voluntary standard does not constitute compliance to any applicable regulatory requirements. Users of the standard are responsible for observing or referring to the applicable laws and regulations.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{1,2}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1872™-2015, IEEE Standard Ontologies for Robotics and Automation.^{3,4}

ISO/IEC 24707:2018, Information technology—Common Logic (CL)—A framework for a family of logic-based languages.⁵

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The IEEE Standards Dictionary Online should be consulted for terms not defined in this clause.⁶ Informative definitions are available in Annex A.

autonomous system: A system, either physically embodied or realized entirely within a software substrate, capable of performing tasks and behaviors with a high degree of autonomy making informed decisions without external direction and with the ability to adapt to changing conditions, knowledge, and constraints.

NOTE—Subclause 4.8 presents further context and motivation for the following three definitions reflecting various legal or technology-oriented preferences regarding norm violations and autonomous systems.

common world view: An ethical autonomous system domain analysis that adopts a middle ground between the Legal World View (LWV) and the Technology World View (TWV) by relying upon the concept of a maturity level of socio-technology governance capabilities to be achieved and certified for governments

¹ The use of the word *must* is deprecated and cannot be used when stating mandatory requirements, *must* is used only to describe unavoidable situations.

² The use of *will* is deprecated and cannot be used when stating mandatory requirements, *will* is only used in statements of fact.

³ IEEE publications are available from the Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁴ The IEEE standards or products referred to in Clause 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

⁵ ISO publications are available from the International Organization for Standardization (<http://www.iso.org/>) and the American National Standards Institute (<http://www.ansi.org/>).

⁶ IEEE Standards Dictionary Online is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

adopting ethically driven robotics and automation systems (ERAS) ontology commitments. The extent and type of responsibility ascriptions that can be ascribed to ethically aware autonomous systems would be based on the level of socio-technology governance achieved.

legal world view: An ethical autonomous system domain analysis predisposition asserting that current and foreseeable future legal systems do not and should not permit ascribing responsibility to autonomous systems for any norm violation, legal or ethical.

technology world view: An ethical autonomous system domain analysis predisposition asserting that emerging advances in artificial intelligence (AI) technology will soon motivate granting autonomous systems with formal and legal agenthood with consequential accountability and responsibility requirements.

3.2 Acronyms and abbreviations

AI	artificial intelligence
CLIF	Common Language Interchange Format
CORA	Core Ontology for Robotics and Automation
CWV	Common World View
DPP	Data Protection and Privacy
ERAS	ethically driven robotics and automation systems
EVM	Ethical Violation Management
LWV	Legal World View
NEP	Norms and Ethical Principles
R&A	robotics and automation
TA	Transparency and Accountability
TLO	top-level ontology
TWV	Technology World View
UML	Unified Modeling Language

4. Ontologies for ethically aligned robotics and automation systems

4.1 Conventions

This document presents the formal definitions for the development of ERAS using Common Logic Interchange Format (CLIF) notation, as defined in ISO/IEC 24707:2018⁷. To facilitate the understanding of the entire document, the ontologies developed were broken into a set of interrelated sub-ontologies. In several cases, the definition of a concept in one of the sub-ontologies is dependent upon a separate concept defined in a different sub-ontology. To make this cross reference clear, this standard uses the following

⁷ Information on references can be found in Clause 2.

notation X:Y to indicate the concept Y is defined at ontology X. For instance, ERAS-TLO:Method makes reference to the concept Method defined in Ontology ERAS-TLO.

This document also uses the shorthand notation proposed by Berardi, Calvanese, and De Giacomo [B11] and by Calvanese and De Giacomo in 2007 [B15] and 2020 [B16] to provide additional semantics to the axioms. As an example, consider the following axiom:

```
(forall (x y) (if (relation x y)
  (and (EntityX x)
    (EntityY y))))
```

It asserts that if the relationship “relation” associated to two instances x and y exists, then x should be an instance of the entity EntityX while y should be an instance of entity EntityY. If the semantics of the relationship entails cardinality constraints such that instance x can be related to some number of p instances of the entity EntityY, then the following shorthand notation is used:

```
(forall (x) (if (EntityX x)
  (>= p ( # { y | (and (relation x y) (EntityY y)) } ))))
```

where # is an operator that is used to indicate the number of elements of a particular set. In this case, $\# \{ y \mid (\text{and}(\text{relation } x \ y) (\text{EntityY } y)) \}$ denotes the number of the elements of the set $\{ y \mid (\text{and}(\text{relation } x \ y) (\text{EntityY } y)) \}$ which is comprised of instances y from entity EntityY that participate in the relationship “relation” with the instance x.

Thus, the previous sentence indicates that the size of set $\{ y \mid (\text{and}(\text{relation } x \ y) (\text{EntityY } y)) \}$ should be greater than or equal to p. This is shorthand notation for the following equivalent CLIF expressions:

```
( >= p ( sizeof ( setof y ( and (relation x y) (EntityY y) )
```

where sizeof and setof are helper operator expressions defined in CLIF. The sizeof operator expression returns the number of elements in a set. The setof operator expression constructs a set of elements bound to a free variable y where each bound value satisfies the logical properties expressed as a conjunction of CLIF operator expressions. More information on this notation can be found in Berardi, Calvanese, and De Giacomo [B11], Calvanese and De Giacomo, 2007 [B15] and Calvanese and De Giacomo, 2020 [B16].

4.2 Background

Due to the complexity of the domain of interest, the Working Group decided to concentrate its efforts on four interrelated domains within the Ethically aligned Robotics and Automation Systems (ERAS) domain, as discussed in Annex B. These domains are as follows:

- *Norms and Ethical Principles (NEP)*: This subdomain formalizes aspects of ethical theories and principles that characterize the norms of expected behaviors for norm aware agents and autonomous systems.
- *Data Protection and Privacy (DPP)*: This subdomain formalizes relevant concepts and relationships characterizing the data protection and privacy rules and regulations that shall be observed and upheld by ethical agents and autonomous systems.
- *Transparency and Accountability (TA)*: This subdomain formalizes the concepts and relationships necessary to enable ethical autonomous systems with capabilities to provide informative explanations for plans and associated action.

- *Ethical Violation Management (EVM)*: This subdomain formalizes concepts and relationships associated with capabilities to detect, assess, and manage ethical violations in autonomous system behavior. In addition to ethical violation conceptualizations, this subdomain also addresses concepts and relationships governing accountability, responsibility, and legal notions of personhood for agents.

For each subdomain, a set of use cases was elaborated to represent relevant and realistic scenarios to help identify the conceptual and relationship terminology for the ontologies. Some examples of use cases can be found in Annex C. This process gave rise to formal models expressed through Unified Modeling Language (UML) diagrams that contain the main concepts and relationships for each subdomain together with a set of axioms written in CLIF to add semantics to the models. These models are presented and axiomatization is elaborated for each aforementioned subdomain in 4.4 through 4.8.

Partitioning the ERAS ontology into four subdomains was one strategy to manage the modeling complexity of the domain of interest. In addition, a graph transformation modeling technique similar to that proposed in Guizzardi, Figueiredo, Hedblom, and Poels [B26] was adopted. This approach utilizes a set of graph transformation rules to simplify fully articulated ontology models that have explicit formalizations for all conceptual categories in the target domain. The set of preconditions contained in each graph transformation rule enables the mapping of categories represented as classes into simpler properties with enumerated value types. Similar rules can be applied to reverse the mapping from enumerations back to categories as classes.

Similarly, this standard manages the complexity of the ERAS domain by representing potential points of extension for elaboration by utilizing category properties with enumerated data types for the range type of the respective properties. The property terms and related enumerated value terms in the vocabulary provide examples of category properties and value sets as candidate points of extension for target domain ontologies that reference the ERAS core ontology conceptualizations. To simplify the semantics of ERAS concepts and relationships, certain class properties are defined with enumerated data types that have informative definitions to denote their intended meaning within the context of their affiliated property class domain. This is as opposed to formalizing each of the enumerated values with individual category classes and axioms.

Concept property terms and associated terms denoting property data types are optional and depict example characterizations of the domain concepts for the respective property terms. In most cases, value terms listed in each enumerated data type have only the corresponding informative definitions to characterize the example semantics and do not possess further formalization in axioms. However, in a few cases, in order to fully express the meaning and commitments for the related domain category of a property, value terms from the property's enumerated data type are referenced in one or more axioms for the category concept. The axioms that formalize the meaning of a Norm derogation process are examples of this latter case. Specifically, these axioms specify the semantics of a Norm derogation process as an Agent Action and its effect on the state of a Norm by referencing value terms from the `norm_state` enumerated data type for the state property of the Norm category.

Target-domain ontologies that require more complex formalizations can apply relevant graph rewriting rules, either manually or automatically with supporting tools, to elaborate the ERAS core vocabulary. The set of informative definitions for the ERAS enumerated value terms are listed in Annex A together with the informative definitions of the concepts identified across all ontologies.

4.3 Top-level definitions

As a core ontology, the Ethically aligned Robotics and Automation Systems (ERAS) ontology represents a mid-level set of formalizations and commitments that are platform independent and intended to fit between an upper top-level or foundational ontology and lower domain and application specific ontologies. However, potential users of the standard may have different requirements regarding top-level ontology alignments. Some may require the commitments of one or more existing top-level ontologies while other

user communities would need only a minimal set of top-level commitments to complete the formalization of the concepts, terms, and commitments axiomatized in the ERAS ontology.

This standard introduces the ERAS top-level concepts that are intended to represent a standalone set of formalizations that complete the definitions and commitments expressed in the four ERAS subdomains. They define a minimal set of terms for that purpose. Users that do not require alignments with other existing foundational ontologies would use them.

4.4 ERAS top-level concepts

The ERAS top-level concepts and relationships are derived as a minimal set of ontological commitments appropriate for completing the formalization of concepts and relationships relevant to the characterization of ethically oriented agents and autonomous systems as identified in the four ERAS subdomains. While this subclause includes a complete set of axioms for the ERAS top-level concepts and axioms, the formalizations are not intended to be applicable as a top-level ontology in other contexts. Many of the concepts are similar to those found in other top-level ontologies such as SUMO (Niles and Pease [B39]), GFO (Herre [B27]), UFO (Guizzardi [B25]), and KR (Sowa [B50]). However, the set of concepts that comprise them is smaller than any of those taxonomies and all of the relevant axioms are expressed in CLIF. Figure 1 depicts the UML model for ERAS top-level concepts.

All individuals are entities:

(forall (x) (Entity x))

Entities are specialized into two subcategories that distinguish between Physical and Abstract conceptualizations.

(forall (x) (if (Physical x) (Entity x)))

(forall (x) (if (Abstract x) (Entity x)))

A Physical entity is an entity that has a location in space-time, i.e., physical entities are spatio-temporal located at a pose in the world at a specific time. Both Time and SpatioTemporalPlace are subcategories of Abstract, i.e.,

(forall (x) (if (SpatioTemporalPlace x) (Abstract x)))

(forall (x) (if (Time x) (Abstract x)))

(forall (x) (if (Physical x)
 (exists (p t)
 (and (SpatioTemporalPlace p)
 (Time t)
 (located_at x p)
 (present_at x t)))))

The two referenced relationships have the following formalizations:

(forall (d r) (if (located_at d r)
 (and (Physical d)
 (SpatioTemporalPlace r))))

(forall (d r) (if (present_at d r)
 (and (Physical d)
 (Time r))))

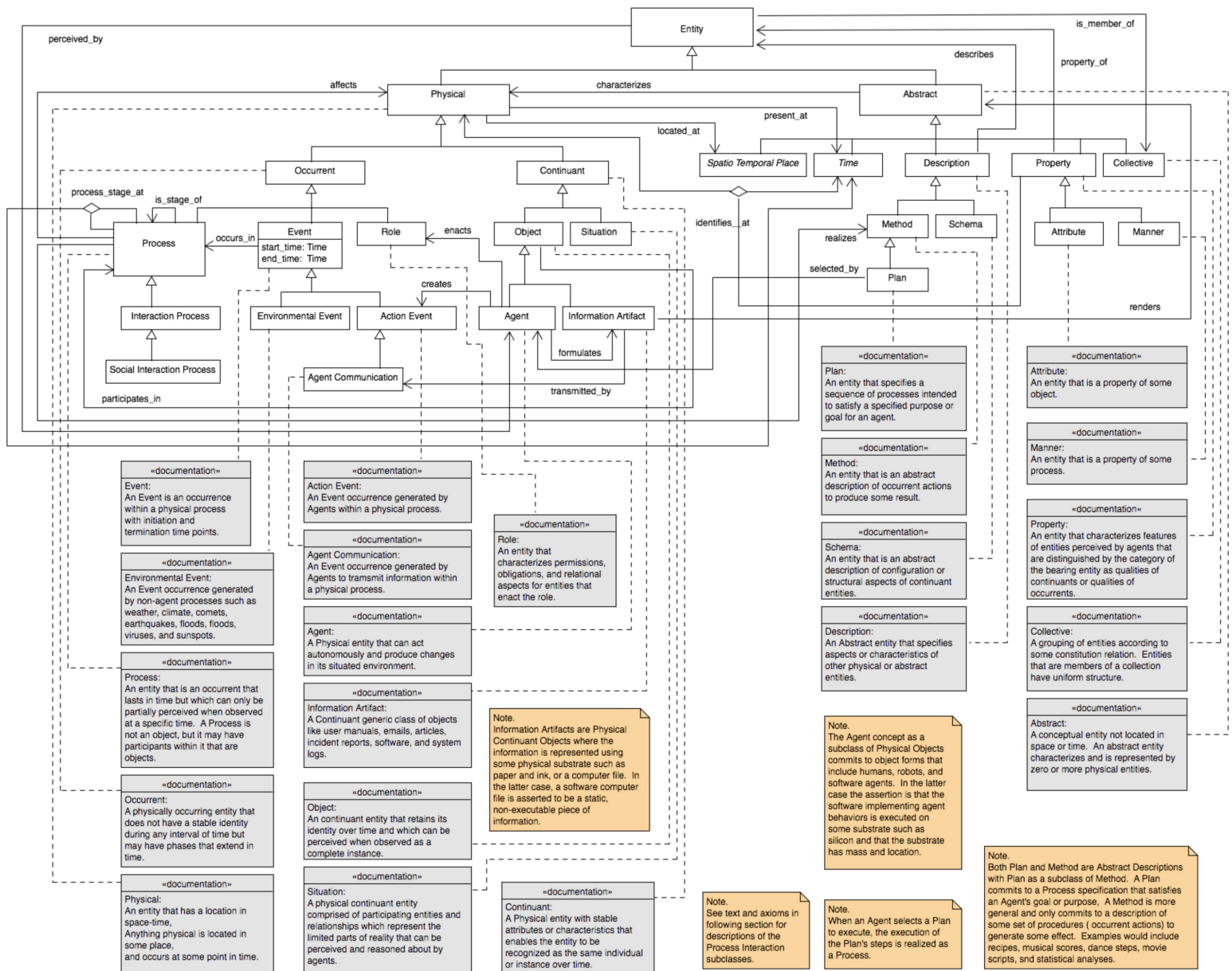


Figure 1 — Ethically driven robotics and automation systems top-level concepts UML diagram

The Abstract category classifies entities that are conceptual only and not located in space or time.

```
(forall (a) (if (Abstract a )  
                (and (Entity a )  
                    (not (Physical a))))))
```

```
(forall (p) (if (Physical p)  
                (and (Entity p )  
                    (not (Abstract p))))))
```

Abstract entities characterize and represent zero or more Physical entities.

```
(forall (x y) (if (characterizes x y)  
                 (and (Abstract x)  
                     (Physical y))))
```

The cardinality of this relationship is represented using shorthand notation for expressing multiplicity constraints on binary associations.

```
(forall (a) (if (Abstract a)  
                (>= 0 ( # { p | (and (characterizes a p) (Physical p)) } ))))
```

The Physical category is specialized into two subcategories that distinguish between Occurrent and Continuant conceptualizations.

```
(forall (x) (if (Occurrent x) (Physical x)))
```

```
(forall (x) (if (Continuant x) (Physical x)))
```

The Occurrent category classifies physically occurring entities that do not have a stable identity during any interval of time. Occurrent entities happen in time and may have temporal parts or phases that extend in time but cannot be wholly perceived at any point in time. Occurrent entities are in states of flux that prevent them from being recognized by a stable set of properties.

```
(forall (o) (if (Occurrent o)  
                (exists(t1 t2)  
                    (and (Time t1)  
                        (Time t2)  
                        (not (= t1 t2))  
                        (present_at o t1)  
                        (present_at o t2)  
                        (not (exists a)  
                            (and (Property a)  
                                (identifies_at a o t1)  
                                (identifies_at a o t2))))))))
```

```
(forall (a p t) (if (identifies_at a p t)  
                   (and (Property a)  
                       (Physical p)  
                       (Time t))))
```

The Occurrent category is specialized into three subcategories to distinguish between concepts of Process, Event, and Role.

```
(forall (x) (if (Process x)(Occurrent x)))
```

(forall (x) (if (Event x) (Occurrent x)))

(forall (x) (if (Role x) (Occurrent x)))

A Process entity is an Occurrent entity that lasts in time but that can only be partially perceived when observed at a specified time. A Process is not an object but may have participants within it that are objects, and it may be perceived in stages, at least one, as it occurs over time.

(forall (p) (if (Process p)
 (exists (t q a)
 (and (Agent a)
 (Time t)
 (Process q)
 (process_stage_at q p t)
 (not (perceived_by p a))
 (perceived_by q a))))))

(forall (s p t) (if (process_stage_at s p t)
 (and (Process s)
 (Process p)
 (is_stage_of s p)
 (Time t))))

(forall (d r) (if (is_stage_of d r)
 (and (Process d)
 (Process r))))

(forall (d r) (if (perceived_by d r)
 (and (Entity d)
 (Agent r))))

A Process may affect a Physical entity.

(forall (p e) (if (affects p e)
 (and (Process p)
 (Physical e))))

A Role entity is an Occurrent entity that characterizes permissions, obligations, and relational aspects for Agents that enact the Role.

(forall (a r) (if (enacts a r)
 (and (Agent a)
 (Role r))))

An Event entity is an Occurrent entity within a physical Process with initiation and termination time points.

(forall (e) (if (Event e)
 (exists (t n p)
 (and (Process p)
 (Time t)
 (Time n)
 (start_time e t)
 (end_time e n)
 (occurs_in e p))))))

(forall (e p) (if (occurs_in e p)

(and (Event e)
(Process p))))

(forall (e t) (if (or (start_time e t)
(end_time e t))
(and (Event e)
(Time t))))

The Event category is specialized into two subcategories: Action Event and Environmental Event.

(forall (x) (if (ActionEvent x) (Event x)))

(forall (x) (if (EnvironmentalEvent x) (Event x)))

(forall (e) (if (EnvironmentalEvent e)
(not (ActionEvent e))))

(forall (e) (if (ActionEvent e)
(not (EnvironmentalEvent e))))

An Action Event entity is an Event occurrence created by Agents within a process.

(forall (e) (if (ActionEvent e)
(exists (a p)
(and (Agent a)
(Process p)
(occurs_in e p)
(creates a e))))))

(forall (a e) (if (creates a e)
(and (Agent a)
(ActionEvent e))))

An Environmental Event entity is an Event occurrence generated by non-agent processes.

(forall (e) (if (EnvironmentalEvent e)
(exists (p)
(and (Process p)
(occurs_in e p)
(not (and (InteractionProcess p)
(ActionEvent e))))))

The Action Event category is specialized with the Agent Communication subcategory.

(forall (x) (if (AgentCommunication x) (ActionEvent x)))

An Agent Communication entity is an Action Event generated by Agents to transmit information within a Physical Process.

(forall (x) (if (AgentCommunication x)
(exists (d)
(and (InformationArtifact d)
(transmitted_by d x))))))

(forall (d r) (if (transmitted_by d r)

(and (InformationArtifact d)
(AgentCommunication r))))

The Process category has Interaction Process as a subcategory specialization.

(forall (i) (if (InteractionProcess i) (Process i)))

An Interaction Process is a Process that includes at least one Action Event that was generated by one or more Agents.

(forall (i) (if (InteractionProcess i)
(exists (a e)
(and (Agent a)
(ActionEvent e)
(creates a e)
(occurs_in e i)
(participates_in a i))))))

(forall (d r) (if (participates_in d r)
(and (Object d)
(Process r))))

The Interaction Process category has Social Interaction Process as a subcategory specialization.

(forall (s) (if (SocialInteractionProcess s) (InteractionProcess s)))

A Social Interaction Process is an Interaction Process that includes multiple Agents engaged in Agent Communication sub process stages.

(forall (s) (if (SocialInteractionProcess s)
(exists (a b c)
(and (Agent a)
(Agent b)
(not (= a b))
(AgentCommunication c)
(participates_in a s)
(participates_in b s)
(occurs_in c s))))))

The Continuant category classifies physical entities with stable attributes or characteristics that enable the entity to be recognized as the same individual or instance over time. This category is specialized into subcategories to distinguish the Object and Situation conceptualizations.

(forall (x) (if (Object x) (Continuant x)))

(forall (x) (if (Situation x) (Continuant x)))

(forall (c) (if (Continuant c)
(exists (a t1 t2)
(and (Attribute a)
(Time t1)
(Time t2)
(not (= t1 t2))
(present_at c t1)
(identifies_at a c t1))

(present_at c t2)
(identifies_at a c t2))))

An Object entity is a Continuant entity that retains its identity over time and which can be perceived when observed as a complete instance. Object entities can have different properties at different times and therefore can undergo change.

(forall (o) (if (Object o)
 (and (Continuant o)
 (not (Situation o))))))

The Situation category is a physical Continuant entity comprised of participating entities and relationships that represent the limited parts of reality that can be perceived and reasoned about by agents.

(forall (s) (if (Situation s)
 (and (Continuant s)
 (not (Object s))))))

The Object category is specialized into the subcategories of Agent and Information Artifact.

(forall (x) (if (Agent x) (Object x)))

(forall (x) (if (InformationArtifact x) (Object x)))

An Agent entity is a Physical entity that can create Action Events to produce changes in the agent's physical situation and which can be perceived by it or other agents. An Agent has a repertoire of Plans that describe Methods for achieving or satisfying agent intentions. The plan methods specify sequences of Process actions that affect changes in the Physical situation of the Agent.

(forall (d r) (if (selected_by d r)
 (and (Plan d)
 (Agent r))))

(forall (a p m e) (if (and (Agent a)
 (Process p)
 (participates_in a p)
 (Plan m)
 (Physical e)
 (not (= e p))
 (realizes p m)
 (affects p e))
 (selected_by m a))))

The Information Artifact category is a Continuant generic class of Objects that renders abstract descriptive ideas, expressions, and facts as tangible artifacts using printed text, electronic media or some form of physical substrate. Information Artifact entities are formulated by Agents. Examples in the ERAS domain include user manuals, system logs, incident reports, articles, emails, and software.

(forall (d r) (if (formulates d r)
 (and (Agent d)
 (InformationArtifact r))))

(forall (d r) (if (renders d r)
 (and (InformationArtifact d)

(Abstract r))))

(forall (x) (if (InformationArtifact x)
 (exists (a)
 (and (Agent a)
 (formulates a x))))))

The Abstract category is specialized into three instantiatable subcategories: Description, Property, and Collective, and two abstract, non-instantiatable subcategories: Time and Spatio Temporal Place.

(forall (x) (if (Description x) (Abstract x)))

(forall (x) (if (Property x) (Abstract x)))

(forall (x) (if (Collective x) (Abstract x)))

(forall (x) (if (Time x) (Abstract x)))

(forall (x) (if (SpatioTemporalPlace x) (Abstract x)))

The non-instantiatable abstract Time category represents a linear sequence of time points. The non-instantiatable abstract Spatio Temporal Place category represents the combinatorial properties of space and time qualities that characterize locations of Occurrent and Continuant Entities. Physical entities are present at or exist at various points in Time and are located at various Spatio Temporal Places.

The Description category classifies entities that specify aspects or characteristics of other physical or abstract entities.

(forall (d) (if (Description d)
 (exists (e)
 (and (Entity e)
 (describes d e))))))

(forall (d e) (if (describes d e)
 (and (Description d)
 (Entity e))))

The Description category is specialized into two subcategories: Method and Schema.

(forall (x) (if (Method x) (Description x)))

(forall (x) (if (Schema x) (Description x)))

The Method category classifies an entity that is an abstract description of Occurrent Process actions to produce some result.

(forall (x) (if (Method x)
 (exists (y)
 (and (Process y)
 (describes x y)
 (realizes y x))))))

(forall (d r) (if (realizes d r)
 (and (Process d)
 (Method r))))

The Method category is specialized into the Plan subcategory.

(forall (x) (if (Plan x) (Method x)))

The Plan category classifies an entity that specifies a sequence of processes intended to satisfy a specified purpose or goal for an Agent by affecting changes in the Agent's Physical situation.

(forall (x p) (if (and (Plan x)
 (Process p)
 (realizes p x))
 (exists (e)
 (and (Physical e)
 (not (= e p))
 (affects p e))))))

The Schema category classifies an entity that is an abstract description of configuration or structural aspects of Continuant entities.

(forall (x) (if (Schema x)
 (exists (c)
 (and (Continuant c)
 (describes x c))))))

The Property category classifies an entity that characterizes features of entities perceived by Agents that are distinguished by the category of the bearing entity as qualities of Continuants or qualities of Occurrents.

(forall (p) (if (Property p)
 (exists (a e)
 (and (Agent a)
 (Entity e)
 (property_of p e)
 (perceived_by e a))))))

The Property category is specialized into two subcategories: Attribute and Manner.

(forall (x) (if (Attribute x) (Property x)))

(forall (x) (if (Manner x) (Property x)))

The Attribute category classifies an entity that is a property of some object, i.e.,

(forall (x) (if (Attribute x)
 (exists (o)
 (and (Object o)
 (property_of x o))))))

The Manner category classifies an entity that is a property of some process, i.e.,

(forall (x) (if (Manner x)
 (exists (p)
 (and (Process p)
 (property_of x p))))))

(forall (d r) (if (property_of d r)
 (or (and (Attribute d)

(Object r))
(and (Manner d)
(Process r))))))

The Collective category classifies entities that are grouped together according to some constitution relation. Entities that are members of a collection have uniform structure.

(forall (d r) (if (is_member_of d r)
(and (Entity d)
(Collective r))))))

(forall (c) (if (Collective c)
(exists (e r)
(and (Entity e)
(is_member_of e c)
(Description r)
(describes r c))))))

4.5 Norms and Ethical Principles

Figure 2 shows the UML diagram that captures the main concepts and relationships elicited during the investigation of the Norms and Ethical Principles (NEP) subdomain. It focuses on aspects of ethical theories and principles that characterize the norms of expected behaviors for norm-oriented agents and autonomous systems. This diagram links some concepts to ones already represented in the ERAS top-level ontology and other related ontologies.

A Norm is a Method entity that describes a set of rules and methods governing behavior expected for norm-aware agents. Norm types are derived from respective ethical theories and are possibly influenced by agent social contexts. An ethical theory is a systematization of concepts specifying or recommending aspects of morally correct behavior based on philosophical values and the characterization of right and wrong conduct. For norm aware agents, normative ethical theory is concerned with the practical means of determining a moral course of action.

(forall (n) (if (Norm n) (ERAS-TLO:Method n)))

(forall (t) (if (EthicalTheory t) (ERAS-TLO:Method t)))

(forall (n) (if (Norm n)
(exists (t s)
(and (EthicalTheory t)
(SocialCollection s)
(specifies_norm_modality t n)
(is_prescribed_by n t)
(influences_norm_applicability s n))))))

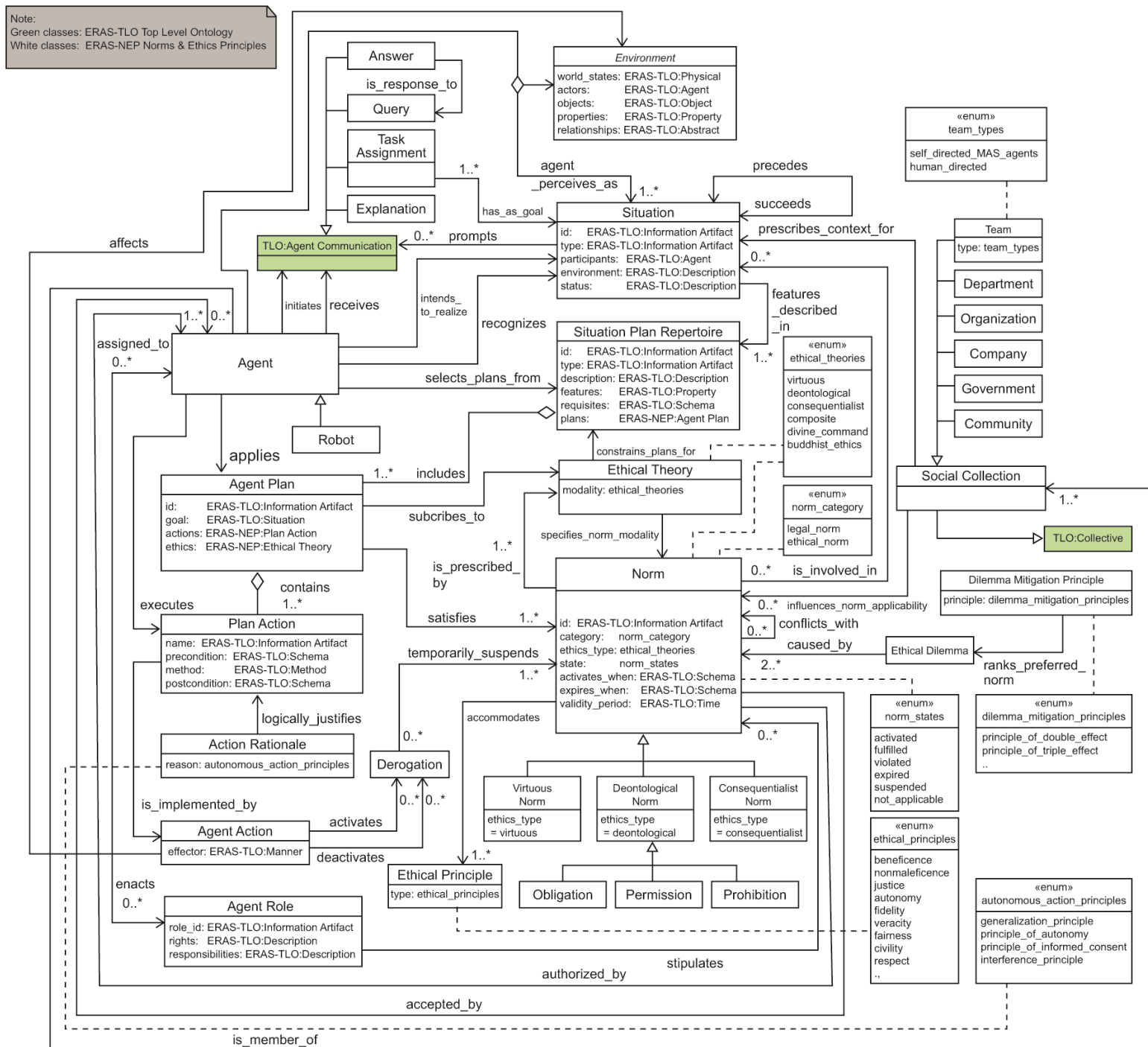


Figure 2 — Norms and Ethical Principles UML diagram

A descriptive characterization of representative forms of Ethical Theories includes the following set of enumerated examples.

```
( = ( Description ethical_theories)
  { virtuous
    deontological
    consequentialist
    composite
    divine_command
    Buddhist_ethics })
```

The various philosophical forms of Ethical Theories aim to specify the modality of Norms intended to guide the behaviors of ethically aware Agents. Each Norm is prescribed by at least one Ethical Theory.

```
(forall (d r) (if (specifies_norm_modality d r)
  (and (EthicalTheory d)
    (Norm r) )))
```

```
(forall (d r) (if (is_prescribed_by d r)
  (and (Norm d)
    (EthicalTheory r) )))
```

Using the shorthand notation,

```
(forall (n) (if (Norm n)
  (>= 1 ( # { t | (and (is_prescribed_by n t) (EthicalTheory t)) } ))))
```

An extensive set of Norms exists for Ethically-aware Agents to consider. For autonomous agents, Norms are authorized by one or more Agents.

```
(forall(d) (if (Norm d)
  (exists(r)
    (and (Agent r)
      (authorized_by d r))))))
```

```
(forall (n a) (if (authorized_by d r)
  (and (Norm d)
    (Agent r))))
```

Using shorthand notation,

```
(forall (n) (if (Norm n)
  (>= 1 ( # { a | (and (authorized_by n r) (Agent a)) } ))))
```

Agents may accept zero or more Norms.

```
(forall (d r) (if (accepted_by d r)
  (and (Norm d)
    (Agent r))))
```

(forall (n) (if (Norm n)
(≥ 0 (# { a | (and (accepted_by n a) (Agent a)) }))))

The Norm category has at least three subcategories representing distinct behavioral courses of actions expressed by the main ethical theories.

(forall (n) (if (VirtuousNorm n) (Norm n)))

(forall (n) (if (DeontologicalNorm n) (Norm n)))

(forall (n) (if (ConsequentialistNorm n) (Norm n)))

A Virtuous Norm is derived from the virtuous ethical theory that elucidates correct action choices based on alignment with certain dispositional character traits or virtues that are appropriate and praiseworthy. From this perspective, correct agent behavior is achieved by adhering to character traits deemed praiseworthy and not blameworthy.

(forall (n) (if (VirtuousNorm n) (= (ethics_type n) virtuous)))

A Deontological Norm is derived from the deontological ethical theory that stipulates correct action choices based on the action's conformity to universal rules for judging rightness or wrongness of an act. From this perspective, correct behavior is independent of the resulting consequences.

(forall (n) (if (DeontologicalNorm n) (= (ethics_type n) deontological)))

A Consequentialist Norm is derived from the Consequentialist ethical theory that elucidates correct action choices based on the consequences that the action produces. Generally, actions that are expected to result in a greater intrinsic good are preferred.

(forall (n) (if (ConsequentialistNorm n) (= (ethics_type n) consequentialist)))

Additionally, Deontological Norms can be further classified as follows:

- *Obligation*: What an agent should do — an attribute that applies to propositions that an agent is required by some authority to make true,
(forall (n) (if (Obligation n) (DeontologicalNorm n))).
- *Permission*: What an agent may do — an attribute that applies to propositions that an agent is permitted, by some authority to make true,
(forall (n) (if (Permission n) (DeontologicalNorm n))).
- *Prohibitions*: What an agent is forbidden to do — an attribute that applies to propositions that an agent is forbidden, by some authority to make true,
(forall (n) (if (Prohibition n) (DeontologicalNorm n))).

Some Norms are specified and enforced as laws within a social legal system. Consequently, the ERAS ontology commits to an enumerated norm type to distinguish between legal norm and ethical norm categories.

(= (Description norm_category)
{ legal_norm
ethical_norm })

Norm methods influence the behaviors of norm-aware Agents by constraining the Agent Plans that Agents select for realizing Agent objectives and goals. In this context, Norm methods have life cycle states. A descriptive characterization of these states includes the following enumerated examples.

```
( = ( Description norm_states )  
  { activated  
    fulfilled  
    violated  
    expired  
    suspended  
    not_applicable } )
```

Norm methods accommodate ethical principles that specify descriptions of general moral proposition and value judgments that characterize and justify particular ethical prescriptions and evaluations of agent actions. The ERAS ontology commits to having each Norm specification accommodate at least one Ethical Principle.

```
(forall (p) (if (EthicalPrinciple p) (ERAS-TLO:Method p)))
```

```
(forall (d r) (if (accommodates d r)  
  (and (Norm d)  
    (EthicalPrinciple r))))
```

```
(forall (n) (if (Norm n)  
  (exists (p)  
    (and (EthicalPrinciple p)  
      (accommodates n p) )))))
```

A descriptive characterization of representative ethical principles includes the following enumerated examples:

```
( = ( Description ethical_principles )  
  { beneficence  
    nonmaleficence  
    justice  
    autonomy  
    fidelity  
    veracity  
    fairness  
    civility  
    respect } )
```

Agent Plans typically need to satisfy more than one Norm and consequently the multiple constraints represented by the requisite Norms may lead to conflicts or Ethical Dilemmas.

An Ethical Dilemma is a decision-making situation arising between conflicting normative rules of behavior in which none of the choices are deemed unambiguously acceptable or preferable. Ethical Dilemmas are caused by, at least, two such conflicting Norms.

```
(forall (d) (if (EthicalDilemma d) (ERAS-TLO:Situation d)))
```

```
(forall (d) (if (EthicalDilemma d)
```

```
(exists (a b)
  (and (Norm a)
    (Norm b)
    (not (= a b))
    (caused_by d a)
    (caused_by d b)
    (or (conflicts_with a b) (conflicts_with b a))))))
```

```
(forall (d r) (if (conflicts_with d r)
  (and (Norm d)
    (Norm r)
    (not (= d r))))))
```

A Norm may conflict with zero or more norms. Using shorthand notation,

```
(forall (n) (if (Norm n)
  (>= 0 (#{ c | (and (conflicts_with n c) (Norm c)) } ))))
```

A Norm involved in a Norm conflict may be one of the norms causing an Ethical Dilemma.

```
(forall (d r) (if (caused_by d r)
  (and (EthicalDilemma d)
    (Norm r)
  )))
```

It takes at least two norms to cause an Ethical Dilemma.

```
(forall (d) (if (EthicalDilemma d)
  (>= 2 (#{ n | (and (caused_by d n) (Norm n)) } ))))
```

Ethical Dilemmas can be resolved by applying various Dilemma Mitigation Principles to rank a preferred Norm among the conflicting Norms.

```
(forall (p) (if (DilemmaMitigationPrinciple p) (ERAS-TLO:Method p)))
```

```
(forall (d r) (if (ranks_preferred_norm d r)
  (and (DilemmaMitigationPrinciple d)
    (EthicalDilemma r)
  )))
```

A descriptive characterization of representative dilemma mitigation principles includes the following enumerated examples:

```
(= (Description dilemma_mitigation_principles)
  { principle_of_double_effect
    principle_of_triple_effect } )
```

Norms impact how Agents act in their situated environment. The ERAS Agent category is defined in the ERAS Top-Level Concepts ontology as a subcategory of a Physical Continuant Object. An Agent entity is an entity that can act on its own and produce changes in its situated environment. The NEP subdomain specializes the ERAS Top-Level Concept with additional relationships.

(forall (x) (if (Agent x) (ERAS-TLO:Agent x)))

The Agent category is specialized with a Robot subcategory, which is intended to be conceptually equivalent to the CORA:Robot concept without requiring the direct importation of the CORA ontology.

(forall (x) (iff (Robot x) (CORA:Robot x)))

Agents perceive and act in an environment. In the context of the ERAS ontology, the Environment category is an abstraction that classifies an external collection of entities, entity properties, entity relationships, and occurrent processes that pose potential internal Agent conceptualizations derived from Agent perceptions of the external entities present in the Environment. The set of entity conceptualizations that are internalized as Agent perceptions is bounded by the type, capabilities, and focus of the sensors employed by the Agent, and consequently may be a subset of the entities present in the environment. This limited entity set represents that portion of an Agent's environmental reality about which it can perceive, interpret, and reason. Note however that the entities present in the Environment are not dependent upon Agent perceptions. Internally, an Agent's perceptions of its external circumstances are represented and classified by the Situation category.

As autonomous actors, Norm-aware Agents perceive, recognize, and become aware of Situations presented in their environments. They respond with the selection and application of appropriate Agent Plans for realizing goals and reacting to recognized Situations.

This ontology subdomain extends the Situation category as described in the ERAS Top-Level Concepts with additional relationships between the Agent and Environment categories.

(forall (e) (if (Environment e) (ERAS-TLO:Continuant e)))

(forall (s) (if (Situation s) (ERAS-TLO:Situation s)))

(forall (s) (if (Situation s)
 (exists (e a)
 (and (Environment e)
 (Agent a)
 (agent_perceives_as a e s)
 (recognizes a s))))))

(forall (a e s) (if (agent_perceives_as a e s)
 (and (Agent a)
 (Environment e)
 (Situation s))))

(forall (d r) (if (recognizes d r)
 (and (Agent d)
 (Situation r))))

(forall (d r) (if (intends_to_realize d r)
 (and (Agent d)
 (Situation r))))

For any environment in which an agent is a participant, the agent formulates at least one situation representation that is its perception of the environmental state. Using shorthand notation,

```
(forall (a e) (if (and (Environment e)
                        (Agent a))
  (>= 1 ( #{ s | (and (agent_perceives_as a e s) (Situation s)) } ))))
```

Situations, as Agent perceptions of environmental circumstances, may involve zero or more Norms.

```
(forall (d r) (if (is_involved_in d r)
  (and (Norm d)
        (Situation r))))
```

```
(forall (n) (if (Norm n)
  (>= 0 ( #{ s | (and (is_involved_in n s) (Situation s)) } ))))
```

As Agents interact with their Environment and sense the world states represented as Situations, their perceptions and intentions generate Situation sequences establishing succeeds and precedes relationships.

```
(forall (d r) (if (or (succeeds d r)
  (precedes d r))
  (and (Situation d)
        (Situation r)
        (not (= d r)))))
```

Agents interact with their environment by selecting Agent Plans from a repertoire of plans that are relevant for the goals and responses determined by Agent reasoning processes. Agent Plans that are included in a Situation Plan Repertoire consist of specifications, partial or complete, for a sequence of agent actions to achieve target goals, objectives, and services to realize agent intentions. Agent Plans as subclasses of ERAS-TLO:InformationArtifact render ERAS-TLO:Abstract Plans into some physical substrate.

```
(forall (p) (if (AgentPlan p) (ERAS-TLO:InformationArtifact p)))
```

```
(forall (x) (if (AgentPlan x)
  (exists (p)
    (and (ERAS-TLO:Plan p)
          (renders x p)))))
```

```
(forall (r) (if (SituationPlanRepertoire r) (ERAS-TLO:InformationArtifact r)))
```

```
(forall (p) (if (AgentPlan p)
  (exists (r t pa aa n)
    (and (SituationPlanRepertoire r)
          (EthicalTheory t)
          (PlanAction pa)
          (AgentAction aa)
          (Norm n)
          (includes r p)
          (subscribes_to p t)
          (constrains_plans_for t r)
          (contains p pa)
          (is_implemented_by pa aa)
          (satisfies p n)))))
```

```
(forall (d r) (if (includes d r)
```

(and (SituationPlanRepertoire d)
(AgentPlan r))))

A Situation Plan Repertoire includes at least one Agent Plan. Using shorthand notation,

(forall (r) (if (SituationPlanRepertoire r)
(≥ 1 (# { p | (and (includes r p) (AgentPlan p)) })))))

Agent Plans included in the Situation Plan Repertoire are constrained by Ethical Theories. And each Agent Plan satisfies one or more Norms that are prescribed by the associated Ethical Theory.

(forall (d r) (if (constrains_plans_for d r)
(and (EthicalTheory d)
(SituationPlanRepertoire r))))

(forall (d r) (if (satisfies d r)
(and (AgentPlan d)
(Norm r))))

(forall (p) (if (AgentPlan p)
(≥ 1 (# { n | (and (satisfies p n) (Norm n)) })))))

(forall (d r) (if (subscribes_to d r)
(and (AgentPlan d)
(EthicalTheory r))))

The commitments of this ontology subdomain asserts that for any Situation that is an Agent's perception of its environment, there exists at least one set of feature descriptions in the Situation Plan Repertoire that characterizes the situation.

(forall (d r) (if (features_described_in d r)
(and (Situation d)
(SituationPlanRepertoire r))
))

(forall (r) (if (SituationPlanRepertoire r)
(≥ 1 (# { s | (and (features_described_in s r) (Situation s)) })))))

This enables an Agent to select relevant Agent Plans from its Situation Plan Repertoire that are qualitatively good matches for circumstances in the current environment situation.

(forall (r a s) (if (and (SituationPlanRepertoire r)
(Agent a)
(Situation s)
(features_described_in s r)
(or (recognizes a s)
(intends_to_realize a s)))
(selects_plans_from a r)))

(forall (d r) (if (selects_plans_from d r)
(and (Agent d)

(SituationPlanRepertoire r))))

(forall (d r) (if (applies d r)
 (and (Agent d)
 (AgentPlan r))))))

Agent Plans contain one or more Plan Actions. A Plan Action, as a constituent of an Agent Plan, specifies the preconditions and postconditions for the application of an agent action to achieve the objectives and goals of the plan. Plan Actions as subclasses of ERAS-TLO:InformationArtifact render ERAS-TLO:Abstract Methods into some physical substrate.

(forall (x) (if (PlanAction x) (ERAS-TLO:InformationArtifact x)))

(forall (x) (if (PlanAction x)
 (exists (m)
 (and (ERAS-TLO:Method m)
 (renders x m))))))

(forall (d r) (if (contains d r)
 (and (AgentPlan d)
 (PlanAction r))))

(forall (p) (if (AgentPlan p)
 (>= 1 (# { a | (and (contains p a) (PlanAction a)) }))))

A Plan Action may have a supporting Action Rationale that logically justifies it as an autonomous action.

(forall (r) (if (ActionRationale r) (ERAS-TLO:Method r)))

(forall (d r) (if (logically_justifies d r)
 (and (ActionRationale d)
 (PlanAction r))))

Reasons for the justification of a Plan Action are based on autonomous action principles (see Hooker and Kim [B29]) similar to conventional principles of reciprocity. A descriptive characterization of such principles includes the following enumerated examples:

(= (Description autonomous_action_principles)
 { generalization_principle
 principle_of_autonomy
 principle_of_informed_consent
 interference_principle })

When Plan Action preconditions hold, the Agent applying the Agent Plan of the constituent Plan Action executes the Plan Action.

(forall (d r) (if (executes d r)
 (and (Agent d)
 (PlanAction r))))

Plan Actions are implemented by Agent Actions. An Agent Action is an operation or effector that implements the method specified in the Plan Action. The Agent Action is applied and executed by the Agent to affect state changes in an Agent's situated environment.

```
(forall (x) (if (AgentAction x) (ERAS-TLO:Process x)))
```

```
(forall (d r) (if (is_implemented_by d r)
  (and (PlanAction d)
    (AgentAction r))))
```

```
(forall (a pa aa) (if (and (Agent a)
  (PlanAction pa)
  (AgentAction aa)
  (is_implemented_by pa aa)
  (executes a pa))
  (exists (e)
    (and (Environment e)
      (affects aa e))))))
```

While applying Agent Plans that satisfy the set of Norms to which the Agent Plan subscribes, Ethical-aware Agents may need to temporarily suspend or derogate a Norm instance. The Derogation category is the Process that an Agent activates for that purpose. When a Norm instance can be resumed, the Agent deactivates the Derogation.

```
(forall (d) (if (Derogation d) (ERAS-TLO:Process d)))
```

```
(forall (d r) (if (temporarily_suspends d r)
  (and (Derogation d)
    (Norm r))))
```

```
(forall (d r) (if (or (activates d r)
  (deactivates d r))
  (and (AgentAction d)
    (Derogation r))))
```

```
(forall (d a) (if (and (Derogation d)
  (AgentAction a)
  (activates a d))
  (exists (n)
    (and (Norm n)
      (temporarily_suspends d n)
      (= (state n) suspended))))))
```

```
(forall (d a n) (if (and (Norm n)
  (Derogation d)
  (AgentAction a)
  (activates a d)
  (temporarily_suspends d n)
  (= (state n) suspended))))
```