



IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)

STANDARDS

IEEE Computer Society

Developed by the
Simulation Interoperability Standards Organization
Standards Activity Committee (SISO SAC)

IEEE Std 1730™-2022
(Revision of IEEE Std 1730-2010)



IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)

Developed by the

**Simulation Interoperability Standards Organization Standards Activity Committee
(SISO SAC)**
of the
IEEE Computer Society

Approved 16 June 2022

IEEE SA Standards Board

THIS IS A REDLINE DOCUMENT PRODUCED BY TECHSTREET, AN ASME COMPANY.

This document is intended to provide users with an indication of changes from one edition to the next. It includes a full-text version of the new document, plus an indication of changes from the previous version.

Redlines are designed to save time and improve efficiencies by using the latest software technology to find and highlight document changes. More professionals are using valuable new technologies like redlines, to help improve outcomes in a fast paced global business world.

Because it may not be technically possible to capture all changes accurately, it is recommended that users consult previous editions as appropriate. In all cases, only the current base version of this publication is to be considered the official document.

This Redline is produced solely by Techstreet. IEEE has not prepared this Redline and is not responsible for its accuracy.

Redline Processing Notes:

1. ~~Red Text~~ - Red strikethrough text denotes deletions.
2. Blue Text - Blue underlined text denotes modifications and additions.

Abstract: The recommended practice for Distributed Simulation Engineering and Execution Process (DSEEP) is described in this document. The DSEEP is intended as a high-level process framework into which the lower-level systems engineering practices native to any distributed simulation user can be easily integrated. Simulation architectures include Distributed Interactive Simulation (DIS), High Level Architecture (HLA), and Test and Training Enabling Architecture (TENA).

Keywords: Distributed Interactive Simulation (DIS), distributed simulation, engineering methodology, high level architecture (HLA), IEEE 1730™, M&S, modeling and simulation, system process, Test and Training Enabling Architecture (TENA)

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

Copyright © 2022 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 14 October 2022. 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 978-1-5044-8823-5 STD25506
Print: ISBN 978-1-5044-8824-2 STDPD25506

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 IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. 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 makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. 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. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

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 subcommittees of the IEEE SA Board of Governors 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 and Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

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

¹Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>.

²Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.

by public authorities and private users, neither IEEE nor its licensors 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](#).³ 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](#).⁴ 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](#).⁵

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.

³Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴Available at: <https://standards.ieee.org/standard/index.html>.

⁵Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

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 draft recommended practice was completed, the Distributed Simulation Engineering and Execution Process Working Group had the following membership:

Robert Lutz, *Chair*
Katherine L. Morse, *Vice Chair*

Jake Borah
Dannie Cutts
Bradford Dillman
David Drake

Michael Heaphy
Jean-Louis Igarza
Mark McCall
William Oates
Greg Quilliam

David Ronnfeldt
George Schleh
Robert Siegfried
Eric Whittington

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

Robert Aiello
Tom Berg
Curtis Blais
Shelly Blake-Plock
David Fuschi

Werner Hoelzl
Murray Little
Robert Lutz
James McCall
Bjorn Moller

Katherine L. Morse
R.K. Rannow
Bartien Sayogo
Simone Youngblood
Yu Yuan

When the IEEE SA Standards Board approved this recommended practice on 16 June 2022, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Edward A. Addy
Ramy Ahmed Fathy
J. Travis Griffith
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick

Johnny Daozhuang Lin
Kevin Lu
Daleep C. Mohla
Andrew Myles
Damir Novosel
Annette D. Reilly
Robby Robson
Jon Walter Rosdahl

Mark Siira
Dorothy V. Stanley
Lei Wang
F. Keith Waters
Karl Weber
Sha Wei
Philip B. Winston
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1730-2022, IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP).

Modeling and simulation (M&S) has long been an enabler of many key activities across a broad spectrum of different user communities. Such activities include the use of M&S to support the training of personnel, the test and evaluation of new systems and capabilities, analysis of logistics issues, and more. At its core, M&S is about the reduction of risk. That is, by providing abstract representations of highly complex systems along with the operational environment in which the system exists, users can gain an understanding of the performance and behavior of the system that would not be achievable otherwise. Thus, the insight gained through careful and thoughtful application of M&S assets provides an invaluable means of making wise decisions about how the system should be designed, developed, and employed.

There is a wide variety of models and simulations in use today. Such tools are generally characterized by their general class (live, virtual, or constructive) and by the level of detail that they support. While most M&S tools are designed as stand-alone applications, recent advances in software and networking technology has led to significantly greater usage of distributed simulation environments. In such environments, the unique functionalities offered by a selected set of disparate models and simulations are combined and linked using high-speed networks and advanced simulation services to create an integrated system that appears to users to be a single, very powerful simulation.

There are many different approaches to building distributed simulation environments that rely on a wide variety of different protocols, middleware products, and data management schemes. The choice of which systems engineering methodology works best in any given situation depends on several factors, such as the class of application (e.g., analysis, test, training), the types of M&S assets being linked together, verification, validation, and accreditation/acceptance and security concerns, cost/schedule constraints, and even the background and experience of the development/integration team. While considerable experimentation within the distributed simulation community has driven the current wide range of modern, innovative distributed simulation development and execution methodologies, and supporting implementation mechanisms in use today, the proliferation of such methodologies has made it more difficult to transition M&S assets (including the people that support them) from one user domain to another. In some cases, this difficulty has become a barrier to interoperability across the environments supported by different user domains employing different methodologies.

One means of addressing these problems is to compare the various approaches to distributed simulation environments and look for the common denominators. In other words, determine if there are certain activities and tasks that are common across distributed simulation communities and capture these within a common, unified process model. A common process would be quite valuable for the following reasons:

- There would be a common basis for comparison of different domain-specific methodologies. In other words, by identifying how the various methodologies are the same, it also identifies how they are different, and thus facilitates the understanding of how and when one methodology should be used versus another.
- It would provide a common framework for different communities to describe their native systems engineering practices. In other words, by describing their engineering practices within the context of the generalized activities and tasks in the common process model, they can better communicate with and understand the approaches used by other communities that use this same process framework, and help them to drive toward commonality in development approaches whenever possible.
- It would provide a common language for building mixed-architecture simulation environments. In other words, in circumstances in which more than one simulation architecture is required to meet all of the technical, cost, and schedule constraints associated with the application, a common process model

can help to bridge the communications barriers that frequently occur between the users of the different simulation architectures and helps to identify and establish the necessary agreements that must be made for such mixed-architecture environments to operate properly.

- It would provide an effective “first read” to people that are new to the distributed simulation community that want to learn and understand the high-level view of the current best practices in the distributed simulation community before immersing themselves in the specific systems engineering approach used within their own user domain.

This document describes the recommended practice for Distributed Simulation Engineering and Execution Process (DSEEP). The purpose of this document is to describe a generalized process for building and executing distributed simulation environments. It is not intended to replace the existing management and systems design/development methodologies of user organizations, but rather to provide a high-level framework for simulation environment construction and execution into which lower-level systems engineering practices native to each individual application area can be easily integrated. In addition, the DSEEP is not intended to be prescriptive, in that it does not specify a “one size fits all” process for all users of distributed simulation. Rather, the DSEEP defines a generic, common-sense systems engineering methodology that can and should be tailored to meet the needs of individual applications.

Although every distributed simulation application requires a basic agreement among all member application representatives as to the systems engineering approach that will be used to develop and execute the simulation environment, there can be significant variability in the degree of formality defined in the chosen process. The primary driver for how much formality is required is the size and complexity of the application. For example, in large complex applications with many distributed member applications, project control requirements generally dictate the need for a rigidly defined, highly structured process to facilitate proper communication and coordination among all team members. In such environments, requirements and associated schedules for the delivery of supporting products are generally very explicit, as is the content and format for documentation of these products. In smaller or less complex applications, a less structured process with fewer constraints on the types, formats, and content of supporting products may be perfectly reasonable and may have certain efficiency advantages as compared to a more formalized process.

Other secondary factors may also drive the process selected for a specific application. For instance, some communities may have documentation requirements that are unique to their application area. In this case, the activities required to produce these products must be accounted for in the overall process. The reuse potential of these and other required products may also influence the nature and formality of the activities that produce them. Another factor is the availability of reusable products and persistent development/integration teams as opportunities for shortcuts, whereby it may be possible to identify and take advantage of a more streamlined, efficient development/integration process. Finally, practical resource constraints (e.g., cost, schedule) may dictate how certain activities are performed and how the associated products are produced and documented.

In summary, it is recognized that the needs and requirements of the distributed simulation community are quite diverse. The DSEEP is offered to the distributed simulation community as a means of identifying today’s best practices for the development and execution of simulation environments, as well as identifying and addressing the various technical and managerial issues associated with implementation of these practices. The DSEEP can and should be tailored as appropriate to address the unique issues, requirements, and practical constraints of each individual application. It is expected that this framework will provide a viable foundation for all distributed simulation applications and will assist the users in defining the specific tasks and activities necessary to support their particular needs.

Contents

1. Overview	11
1.1 Scope	11
1.2 Purpose	11
1.3 Word usage	11
2. Normative references	12
3. Definitions, acronyms, and abbreviations	12
3.1 Definitions	12
3.2 Acronyms and abbreviations	13
4. DSEEP: Top-level process flow view	13
5. DSEEP: Detailed product flow view	15
5.1 Overview	15
5.2 Step 1: Define simulation environment objectives	19
5.3 Step 2: Perform conceptual analysis	23
5.4 Step 3: Design simulation environment	27
5.5 Step 4: Develop simulation environment	33
5.6 Step 5: Integrate and test simulation environment	38
5.7 Step 6: Execute simulation	42
5.8 Step 7: Analyze data and evaluate results	44
6. Conclusion	46
Annex A (normative) High Level Architecture process overlay to the DSEEP	48
Annex B (normative) Distributed Interactive Simulation process overlay to the DSEEP	53
Annex C (normative) Tested and Training Enabling Architecture process overlay to the DSEEP	62
Annex D (informative) Bibliography	74

IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)

~~IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.~~

~~This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.~~

1. Overview

1.1 Scope

This recommended practice defines the processes and procedures that should be followed by users of distributed simulations to develop and execute their simulations. It is intended as a higher-level framework into which low-level management and systems engineering practices native to user organizations can be integrated and tailored for specific uses.

1.2 Purpose

The purpose of this standard is to reconcile the process descriptions of multiple simulation communities developed in support of specific projects (e.g., Distributed Interactive Simulation [DIS], Hig Level Architecture [HLA], Test and Training Enabling Architecture [TENA]) to help ensure that the Distributed Simulation Engineering and Execution Process (DSEEP) continues to provide the high-level framework from which all other domain-specific simulation development processes can be derived.

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*).^{6,7}

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.

There are no normative references in this recommended practice.

⁶The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁷The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

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](http://dictionary.ieee.org/) ~~:- Glossary of Terms & Definitions~~ should be consulted for terms not defined in this clause.^{8 1}

conceptual model: An abstraction of what is intended to be represented within a simulation environment, which serves as a frame of reference for communicating simulation-neutral views of important entities and their key actions and interactions. The conceptual model describes what the simulation environment will represent, the assumptions limiting those representations, and other capabilities needed to satisfy the user's requirements. Conceptual models are bridges between the real world, requirements, and design.

member application: An application that is serving some defined role within a simulation environment. This can include live, virtual, or constructive simulation assets, or can be supporting utility programs such as data loggers or visualization tools.

objective: The desired goals and results of the activity to be conducted in the distributed simulation environment expressed in terms relevant to the organization(s) involved.

repository: A central place where modeling and simulation (M&S) resources may be cataloged, stored, or accessed.

requirement: A statement identifying an unambiguous and testable characteristic, constraint, process, or product of an intended simulation environment.

^{8 1} ~~The IEEE Standards Dictionary~~ [Online](http://dictionary.ieee.org/) ~~:- Glossary of Terms & Definitions~~ 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.](#)

scenario: (A) Description of an exercise. It is part of the session database that configures the units and platforms and places them in specific locations with specific missions ([see Tucker \[B15\].^{9, 10}](#)).
(B) An initial set of conditions and timeline of significant events imposed on trainees or systems to achieve exercise objectives.

simulation data exchange model (SDEM): A specification defining the information exchanged at runtime to achieve a given set of simulation objectives. This includes class relationships, data structures, parameters, and other relevant information.

simulation environment: A named set of member applications along with a common simulation data exchange model (SDEM) and set of agreements that are used as a whole to achieve some specific objective.

~~NOTE—For definition (A), see Tucker [B4].^{2, 3}~~

3.2 Acronyms and abbreviations

BOM	base object model
CASE	computer aided software engineering
COTS	commercial off-the-shelf
CPU	central processing unit
DIS	Distributed Interactive Simulation
DSEEP	Distributed Simulation Engineering and Execution Process
FOM	federation object model

^{9, 2}The numbers in brackets correspond to those of the bibliography in Annex [E D](#).

^{10, 3}Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

GOTS	government off-the-shelf
HLA	High Level Architecture
IEEE	Institute for Electrical and Electronics Engineers
LAN	local area network
LROM	logical range object model
M&S	modeling and simulation
MOE	measure of effectiveness
MOP	measure of performance
PDU	protocol data unit
RTI	runtime infrastructure
SDEM	simulation data exchange model
SOM	simulation object model
TENA	Test and Training Enabling Architecture
VV&A	verification, validation, and accreditation/acceptance
WAN	wide area network

4. DSEEP: Top-level process flow view

Because of the great diversity among the needs of the distributed simulation user community, there is a recognized desire to avoid mandating unnecessary constraints on how such environments are developed and executed, and generally to provide a high degree of flexibility ~~with regard~~ ~~to~~ how supporting processes are structured. For instance, the types and sequence of low-level activities required to develop and execute distributed analysis-oriented simulation environments is likely to be quite different from those required to develop and execute distributed training exercises. However, at a more abstract level, it is possible to identify a sequence of seven ~~very~~ basic steps that all distributed simulation applications will need to follow to develop and execute their simulation environments. Figure 1 illustrates each of these steps and is summarized by the following below:

- *Step 1: Define simulation environment objectives.* The user, the sponsor, and the development/integration team define and agree on a set of objectives, and document what needs to ~~must~~ be accomplished to achieve those objectives.
- *Step 2: Perform conceptual analysis.* The development/integration team performs scenario development and conceptual modeling, and develops the simulation environment requirements based upon the characteristics of the problem space.
- *Step 3: Design simulation environment.* Existing member applications that are suitable for reuse are identified, and design activities for member application modifications and/or new member applications are performed. Required functionalities are allocated to the member application representatives, and a plan is produced ~~developed~~ for the development and implementation of the simulation environment.
- *Step 4: Develop simulation environment.* The simulation data exchange model (SDEM) is developed, simulation environment agreements are established, and new member applications and/or modifications to existing member applications are implemented.
- *Step 5: Integrate and test simulation environment.* Integration activities are performed, and testing is conducted to verify that interoperability requirements are being met.

- *Step 6: Execute simulation.* The simulation is executed and the output data from the execution is preprocessed.
- *Step 7: Analyze data and evaluate results.* The output data from the execution is analyzed and evaluated, and results are reported back to the user/sponsor.

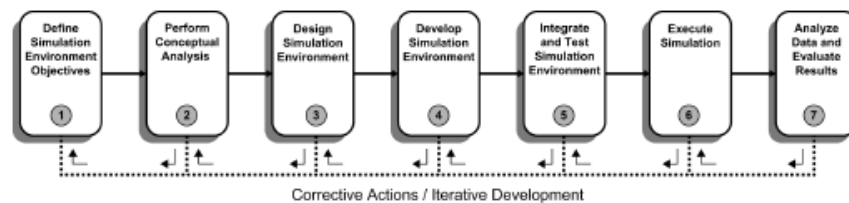


Figure 1—Distributed Simulation Engineering and Execution Process (DSEEP), top-level process flow view

Since this seven-step process can be implemented in [a number of](#) ~~many~~ different ways depending on the nature of the application, it follows that the time and effort required to build and execute a simulation environment can also vary significantly. For instance, it may take a development/integration team several iterations spanning several weeks to months to fully define the real-world domain of interest for large, complex applications. In smaller, relatively simple applications, the same activity could potentially be conducted in a day or less. Differences in the degree of formality desired in the process can also lead to varying requirements for supporting resources.

Within any step of the DSEEP, it may become apparent that the simulation approach is unlikely to ultimately provide sufficiently reliable output data for meaningful analysis to be carried out and therefore that the process should be prematurely terminated. There could be many reasons for this, for example, cost, timescales, lack of viable revised objectives, or non-availability of appropriate member applications. In order to identify such a situation as soon as possible, each major step of the DSEEP should include an assessment as to whether the process should continue or be terminated. Any reason for early termination should be clearly documented.

Personnel requirements can also vary greatly depending on the scope of the distributed simulation environment. In some situations, highly integrated teams composed of several individuals may be needed to perform a single role in a large, complex simulation environment, while a single individual may perform multiple roles in smaller simulation environments. Each distributed simulation user will define their roles according to the needs of their application, the domain they support, and the function performed. Some roles are unique to a single activity in the process, while others are more pervasive throughout the process. Since the applicability of a given role (as well as the set of activities it spans) varies from simulation environment to simulation environment, the activities described in this document specify the roles of individuals only in generic terms. However, best practices recognize some roles that will be prevalent in all distributed simulation activities, regardless of their titles, as follows:

- **User/sponsor:** The person, agency, or organization who determines the need and scope of a distributed simulation exercise or event, and establishes the funding and other required resources. The user/_sponsor also approves the participants, objectives, requirements, and specifications. The user/_sponsor appoints the simulation environment manager and verification, validation, and accreditation/_acceptance (VV&A) agents.
- **Simulation environment manager:** The person responsible for creating the simulation environment, executing the event(s) in the simulation environment, and conducting the post-event activities. The simulation environment manager coordinates with the VV&A agent(s) during these tasks and then reports the results of the event(s) to the user/sponsor.
- **Development/integration team:** The team developing a simulation environment, integrating member applications and systems into the simulation environment, planning for the all aspects of the process, and responsible for ensuring simulations are compliant with simulation environment agreements.
- **Verification and validation (V&V) agent(s):** The person, agency, or organization responsible for verifying and validating member applications or simulation environment.

- **Accreditation/acceptance agent(s):** The person, agency, or organization that accredits member applications or simulation environments for use and reuse for specific purposes or categories of purposes; responsible for certifying that a simulation environment has been verified and validated; and authorizes the use of the simulation environment for its intended use.

A major source of variation in how the seven-step process is implemented relates to the degree of reuse of existing products. In some cases, no previous work may exist; therefore, a new simulation environment may need to be developed based on the defined requirements. In other cases, user communities with established, persistent simulation environments will receive additional requirements. In this circumstance, the users can choose to reuse previous work (e.g., member applications, SDEMs, planning documents) either in part or whole, along with the products of new developmental activities. When an appropriate management structure exists to facilitate this type of development environment, significant savings can be achieved in both cost and development time.

The remainder of this document describes a structured, systems engineering approach to simulation environment development and execution known as the Distributed Simulation Engineering and Execution Process (DSEEP). The seven-step process provides a top-level, process-flow view of the major steps of DSEEP. In the detailed product-flow view, each step is further decomposed into a set of interrelated lower-level activities and supporting information resources. Since the needs of distributed simulation users range from “first use” applications to experienced users, the DSEEP makes no assumptions about the existence of an established core set of member applications or the up-front availability of reusable products. Further, there are no assumptions about the size of the simulation environment, as even users and developers of stand-alone (non-distributed) simulations can benefit from the guidance provided in this document. Although the intention is to define a comprehensive, generalized framework for the construction and execution of distributed simulation environments, it is important to recognize that users of this process model will normally need to adjust and modify the DSEEP as appropriate to address the unique requirements and constraints of their particular application area.

5. DSEEP: Detailed product flow view

5.1 Overview

The DSEEP describes a high-level framework for the development and execution of distributed simulation environments. The intent of the DSEEP is to specify a set of guidelines for the development and execution of these environments that stakeholders can leverage to achieve the needs of their application.

A detailed, product-flow view of the DSEEP is provided in Figure 2. This view illustrates the flow of information across the seven process steps identified in Figure 1. Data flow diagram notation is used in Figure 2 and throughout this document to represent activities (rounded rectangles), data stores (cylinders), and information flows and products (arrows) (see Scrudder et al. [B10 B+]).

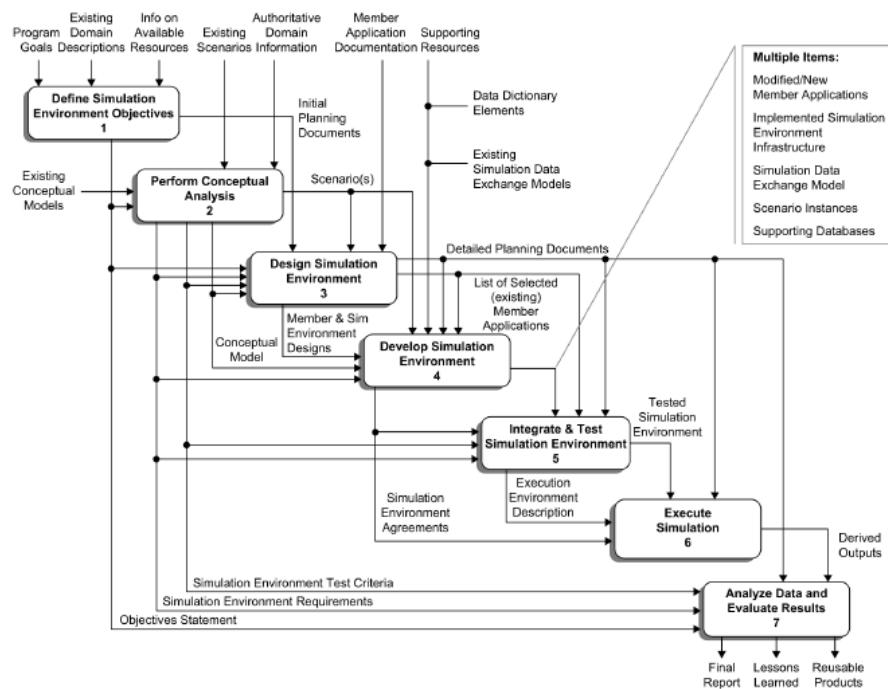


Figure 2—Distributed Simulation Engineering and Execution Process (DSEEP), detailed product flow view

The following subclauses describe the lower-level activities associated with each of the seven major development and execution steps. A tabular view of the activities inherent to each major step is provided in Table 1. Activity descriptions and potential inputs and outputs for the activity and a representative list of recommended tasks are provided. Graphical illustrations of the interrelationships among the activities within each step are also provided. Whenever outputs from one DSEEP activity represent a major input to one or more other activities, the arrow labels explicitly identify the activities that use these outputs. The arrow labels also identify the activities that produce inputs. However, there is a presumption embodied within the DSEEP that once a DSEEP product has been created, it will be available for all subsequent activities, even though the product may not be identified as an input in the activity description. Additionally, once a product is developed, the product may be modified or updated by subsequent activities without such modifications being explicitly identified as either ~~as~~ a task or output. This is the natural cause and effect of iterative and incremental development, which is supported by the DSEEP. Input and output arrows without activity number labels are those in which the information originates from outside or is used outside the scope of the DSEEP.

Table 1—Tabular view of the DSEEP

Step	(1) Define simulation environ- ment objectives	(2) Perform conceptual analysis	(3) Design simulation environment	(4) Develop simulation environment	(5) Integrate and test simulation environment	(6) Execute simulation	(7) Analyze data and evaluate results
Activities	Identify user/ sponsor needs Develop objectives Conduct initial planning	Develop scenario Develop conceptual model Develop simulation environment requirements	Select member applications Design simulation environment Prepare detailed plan	Develop simulation data exchange model Establish simulation environment agreements Implement member application designs Implement simulation environment infrastructure	Plan execution Integrate simulation environment Test simulation environment	Execute simulation Prepare simulation environment outputs	Analyze data Evaluate and feedback results

Although many of the activities represented in the DSEEP diagram and tabular view appear highly sequential, the intention is not to suggest a strict waterfall approach to development and execution. Rather, this process illustration is simply intended to highlight the major activities that occur during development and execution and approximately when such activities are first initiated relative to other development activities. In fact, experience has shown that many of the activities shown in Figure 2 as sequential are actually cyclic and/or concurrent, as was indicated earlier in Figure 1 via the dotted feedback arrows. In general, the lower-level activities will be conducted in the order implied by the information flow between these activities, but the actual timing of these activities is not absolute and is dependent upon the systems engineering practices ~~actually~~ being used.

In addition to the DSEEP products, there are recommended simulation management planning areas for which persistent documentation is key when using incremental development. [The following is](#) a summarized list of these planning areas and resulting documents that are recommended ~~follows~~:

- *Development and execution planning.* The resulting document from this activity identifies the overall plan of action and milestones for the simulation development and execution. There can be multiple views created to depict the schedule of activities, including detailed task and milestone identification for a particular simulation build.
- *Verification and validation (V&V) planning.* The resulting document from this activity identifies the methodology and guidelines for verifying and validating the simulation environment.
- *Test planning.* The resulting document from this activity identifies the methodology and guidelines for executing and evaluating the simulation environment via a formal test. It should include the test criteria imposed on the simulation environment as well as specific test requirements for participating member applications.
- *Configuration management planning.* The resulting document from this activity identifies the methodology and guidelines for establishing and managing configuration baselines. It should indicate the process by which design changes to the simulation environment can be made and the version control utilized by participating member applications.

- *Security planning.* The resulting document from this activity identifies the level of security imposed on the development and execution of the simulation environment. It should also identify the specific security requirements required by participating member applications and possible designated approval authority required by participating member applications.
- *Integration planning.* The resulting document from this activity identifies the methodology and guidance for integration throughout the simulation development.
- *Data management planning.* The resulting document from this activity identifies the strategy for data collection, management, and analysis of the simulation environment as well as the member applications.

The plan information identified above may be captured in distinct documents or folded into other documents that are being mapped in the process, and potentially stored in appropriate repositories. It should be noted that more than one individual may be required to populate and mature these documents as there may be specific program ~~expertise~~ or simulation expertise required. A suggestion for management of the planning documentation is to nominate individuals to maintain and update the documents on a periodic basis or as new information is obtained.

Throughout the DSEEP, there will be areas in which additional information could provide useful guidance in the management, maintenance, and development of a simulation environment. This information should be produced as required in a format that complements the documentation structure being used in the specific process implementation. The information can be captured in distinct documents or folded into other documents that are being managed in the process. Areas in which additional information may be captured include the following:

- *Management tools.* Identify what management tools have been selected to support the DSEEP activities (such as scenario development, requirements, conceptual analysis, VV&A, ~~and~~ configuration management) and how they are being utilized.
- *Reusable products.* Identify the reusable products that have been produced or consumed in the DSEEP activities, such as designs, specifications, source code, documentation, test suites, manuals, procedures, etc.

- *Simulation environment support tools.* Identify what support tools have been selected for the simulation environment (such as data reduction tools, visualization, ~~and~~ simulation environment manager) and how those tools are being utilized in the simulation environment.

Users of the DSEEP should be aware that the activities described in this document, while being generally applicable to most simulation environments, are intended to be tailored to meet the needs of each individual application. For example, DSEEP users should not feel constrained by the products explicitly identified in this document, but rather should produce whatever additional documentation is necessary to support their application. Specific ~~tailoring~~ ~~tailorings~~ of the DSEEP based on the native systems engineering processes inherent to certain widely-used distributed simulation architectures (Distributed Interactive Simulation [DIS], High Level Architecture [HLA], and Test and Training Enabling Architecture [TENA]) are included as annexes to this document. In general, the recommended practices provided in this document should be used as a starting point for developing the most appropriate approach to development and execution for the intended application.

In the subclauses that follow, the description of activities for each DSEEP step is provided and includes a supporting data flow diagram. Each data flow diagram identifies at least one individual activity that is labeled by a number designation (X~~z~~-Y) to show traceability between the step (X) and the activity (Y) and, in the seven-step process, to which the activity is associated. The activity number in these diagrams is intended primarily as an identifier and does not prescribe a particular ordering required for completing a DSEEP step.

5.2 Step 1: Define simulation environment objectives

The purpose of Step 1 of the DSEEP is to define and document a set of needs that are to be addressed through the development and execution of a simulation environment~~z~~ and to transform these needs into a more detailed list of specific objectives for that environment.

Figure 3 illustrates the key activities in this step of the DSEEP. The subclauses that follow describe each of these activities in detail.

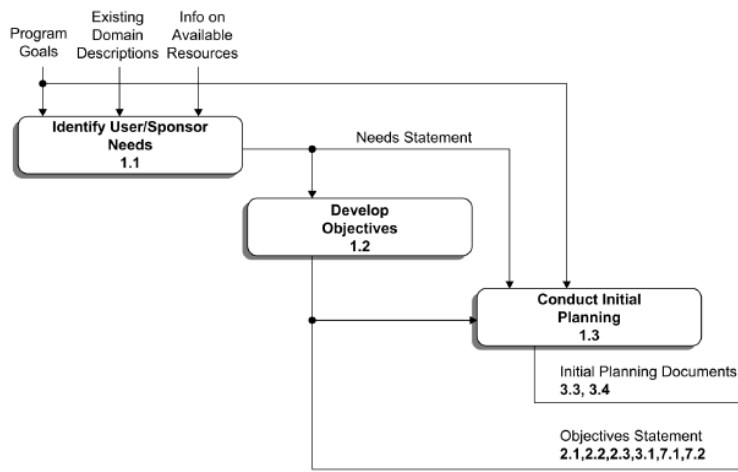


Figure 3—Define simulation environment objectives (Step 1)

5.2.1 Activity 1.1—Identify user/sponsor needs

The primary purpose of this activity is to develop a clear understanding of the problem to be addressed by the simulation environment. The needs statement may vary widely in terms of scope and degree of formalization. It should include, at a minimum, high-level descriptions of critical systems of interest, initial estimates of required fidelity and required behaviors for simulated entities, key events and environmental conditions that need to ~~must~~ be represented in the scenario, and output data requirements. In addition, the needs statement should indicate the resources that will be available to support the simulation environment (e.g., funding, personnel, tools, facilities) and any known constraints that may affect how the simulation environment is developed (e.g., required member applications, due dates, site requirements, security requirements). In general, the needs statement should include as much detail and specific information as is possible at this early stage of the DSEEP.

An explicit and unambiguous statement of user/sponsor needs is critical to achieving clear communication of intent among the development/integration team responsible for the simulation environment. Failure to establish a common understanding of the required product can result in costly rework in later stages of the DSEEP.

5.2.1.1 Activity inputs

Neither this list of inputs, nor any subsequent lists of inputs, is meant to be completely exhaustive, nor are all mandatory for all simulation environments. The potential inputs to this activity include the following:

- Program goals
- Existing domain descriptions
- Information on available resources

5.2.1.2 Recommended tasks

The potential tasks for this activity include the following:

- Analyze the program goals to identify the specific purpose and objective(s) that motivate development and execution of a simulation environment.
- Identify available resources and known development and execution constraints.
- Document the information listed above in a needs statement.

Neither this list of tasks, nor any subsequent lists of tasks, is meant to be completely exhaustive, nor are all mandatory for all simulation environments.

5.2.1.3 Activity outcomes

Neither this list of outcomes, nor any subsequent lists of outcomes, is meant to be completely exhaustive, nor are all mandatory for all simulation environments. The potential outcomes for this activity include the following:

- Needs statement, including:
- Purpose of the simulation environment

- Identified needs (e.g., domain area/issue descriptions, high-level descriptions of critical systems of interest, initial estimates of required fidelity, ~~and~~ required behaviors for simulated entities)
- Key events that need to ~~must~~ be represented in a scenario
- Output data requirements
- Resources that will be available to support the simulation environment (e.g., funding, personnel, tools, facilities)
- Any known constraints that may affect how the simulation environment is developed and executed (e.g., due dates, security requirements)

5.2.2 Activity 1.2—Develop objectives

The purpose of this activity is to refine the needs statement into a more detailed set of specific objectives for the simulation environment. The objectives statement is intended as a foundation for generating explicit simulation requirements (i.e., translating high-level user/sponsor expectations into more concrete, measurable goals). This activity requires close collaboration between the user/sponsor of the simulation environment and the development/integration team to verify that the original needs statement is properly analyzed and interpreted correctly, and that the resulting objectives are consistent with the stated needs.

Early assessments of feasibility and risk should also be performed as part of this activity. In particular, certain objectives may not be achievable given practical constraints (such as cost, schedule, and availability of personnel or facilities) or even limitations on the state-of-the-art of needed technology. Early identification of such issues and consideration of these limitations and constraints in the *objectives statement* will set appropriate expectations for the development and execution effort.

5.2.2.1 Activity inputs

Potential inputs to this activity include the following:

- Needs statement

5.2.2.2 Recommended tasks

Potential tasks for this activity include the following:

- Analyze the needs statement.
- Define and document a prioritized set of objectives, consistent with the needs statement.
- Assess feasibility and risk, and incorporate into objectives statement.
- Develop metrics for each objective.
- Meet with the sponsor to review the objectives and reconcile any differences.
- Begin planning discussions for the development and execution of the simulation environment.

5.2.2.3 Activity outcomes

Potential outcomes for this activity include the following:

- Objectives statement, including:
 - Potential solution approaches and rationale for the selection of a simulation environment as the best approach
 - A prioritized list of measurable objectives for the simulation environment
 - A high-level description of key simulation environment characteristics (repeatability, portability, time management approach, availability, etc.)

- Domain context constraints or preferences, including object actions/relationships and natural environment representations (e.g., geographical regions covered, climate)
- Identification of execution constraints to include functional (e.g., execution control mechanisms), technical (e.g., site, computational and network operations, health/performance monitoring capabilities), economic (e.g., available funding), and political (e.g., organizational responsibilities)
- Identification of security needs and potential security risks, including probable security level and possible designated approval authority (or authorities, if a single individual is not possible)
- Identification of key evaluation measures to be applied to the simulation environment

5.2.3 Activity 1.3—Conduct initial planning

The purpose of this activity is to establish a preliminary simulation environment development and execution plan. The intent is to translate the objectives statement, along with the associated risk and feasibility assessments, into an initial plan with sufficient detail to effectively guide early design activities. The plan may effectively include multiple plans⁷ and should cover such considerations as verification and validation (V&V), configuration management, and security. The plan should also address supporting tools for early DSEEP activities, based on factors such as availability, cost, applicability to the given application, ability to exchange data with other tools, and the personal preferences of the development/integration team.

The plan should also define a high-level schedule of key development and execution events⁷ and provide additional scheduling detail for all pre-development (i.e., prior to Step 4) activities. Note that the initial plan will be updated and extended as appropriate in subsequent development phases as additional information is accumulated throughout the evolution of the distributed simulation design (see [5.4.3.3](#), Activity 3.3).

5.2.3.1 Activity inputs

Potential inputs to this activity include the following:

- Program goals
- Objectives statement

5.2.3.2 Recommended tasks

Potential tasks for this activity include the following:

- Define and document an initial simulation environment development and execution plan, V&V plan, test plan, configuration management plan, security plan, integration plan, and data management plan.
- Identify potential tools to support the initial plan.

5.2.3.3 Activity outcomes

Potential outcomes for this activity include the following:

- Initial planning documents, including:
 - Development and execution plan, including known schedule of activities, detailed [tasks](#), ~~task~~ and identified milestones
 - V&V plan
 - Test plan
 - Configuration management plan
 - Security plan
 - Integration plan

- Data management plan
- Management tools to support scenario development, conceptual analysis, requirements, VV&A, and configuration management
- Reusable products to facilitate reuse across DSEEP activities including supporting the development and update of designs, specifications, source code, documentation, test suites, manuals ~~as well as~~ and procedures, design, ~~and~~ specification
- Simulation environment support tools to support data reduction, visualization, and simulation environment management

5.3 Step 2: Perform conceptual analysis

The purpose of this step of the DSEEP is to develop an appropriate representation of the real-world domain that applies to the defined problem space and to develop the appropriate scenario. It is also in this step that the objectives for the simulation environment are transformed into a set of highly specific requirements that will be used in during design, development, testing, execution, and evaluation.

Figure 4 illustrates the key activities in this step of the DSEEP. The subclauses that follow describe each of these activities in detail.

5.3.1 Activity 2.1—Develop scenario

The purpose of this activity is to develop a functional specification of the scenario. Depending on the needs of the simulation environment, the scenario may ~~actually~~ include multiple scenarios, each consisting of one or more temporally ordered sets of events and behaviors (i.e., vignettes). The primary input to this activity is the domain constraints specified in the objectives statement (Step 1), although existing scenario databases may also provide a reusable starting point for scenario development. Any additional input is provided by the conceptual model, which may be developed in parallel with the scenario. Where appropriate, authoritative sources for descriptions of major entities and their capabilities, behavior, and relationships should be identified prior to scenario construction. A scenario includes the types and numbers of major entities that need to ~~must~~ be represented within the simulation environment, a functional description of the capabilities, behavior, and relationships between these major entities over time, and a specification of relevant environmental conditions (such as urban terrain versus natural area, type of terrain, day/night, climate, etc.) that impact or are impacted by entities in the simulation environment. Initial conditions (e.g., geographical positions for physical objects), termination conditions, and specific geographic regions should also be provided. The product of this activity is a scenario or set of scenarios, which provides a bounding mechanism for conceptual modeling activities.

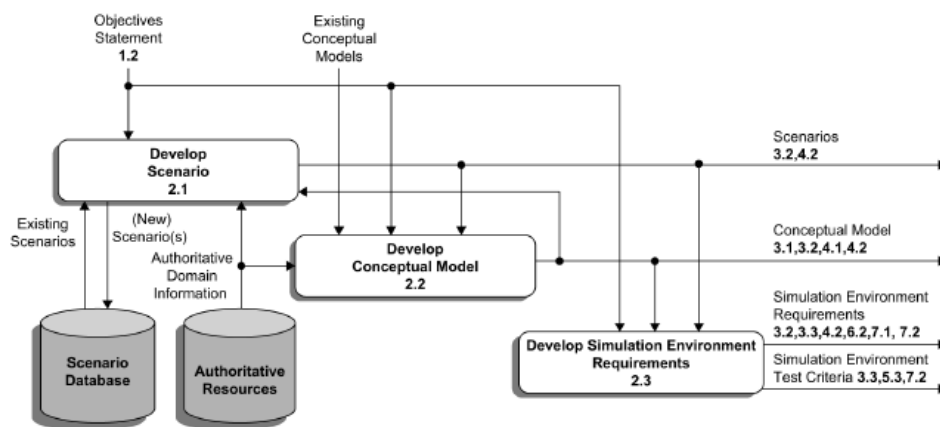


Figure 4—Perform conceptual analysis (Step 2)

The presentation style used during scenario construction is at the discretion of the simulation environment development/integration team. Textual scenario descriptions, event-trace diagrams, and graphical illustrations of geographical positions for physical objects and communication paths all represent effective means of conveying scenario information. Software tools that support scenario development can generally be configured to produce these presentation forms. Reuse of existing scenario databases may also facilitate the scenario development activity.

5.3.1.1 Activity inputs

Potential inputs to this activity include the following:

- Objectives statement
- Existing scenarios
- Conceptual model(s)
- Authoritative domain specific documentation

5.3.1.2 Recommended tasks

Potential tasks for this activity include the following:

- Choose the appropriate tool/technique for development and documentation of the scenario(s).
- Using authoritative domain specific documentation, identify the entities, behaviors, and events that need to be represented in the scenario(s).
- Define one or more representative vignettes that, once executed, will produce the data necessary to achieve the objectives of the simulation environment.
- Define geographic areas of interest.
- Define environmental conditions of interest.

- Define initial conditions and termination conditions for the scenario(s).
- Select an appropriate scenario (or scenario set), or if new scenario information is to be developed, check with the stakeholder that the new scenario(s) will be acceptable.

5.3.1.3 Activity outcomes

Potential outcomes for this activity (in addition to potential planning document updates) include the following:

- Scenario(s), including:
 - Types and numbers of major entities/objects that need to ~~must~~ be represented within the simulation environment
 - Description of entity/object capabilities, behaviors, and relationships
 - Event timelines
 - Geographical region(s)
 - Natural environment condition(s)
 - Initial conditions
 - Termination conditions