IEEE STANDARDS ASSOCIATION
IEEE

IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO)

IEEE Computer Society

Sponsored by the
Simulation Interoperability Standards Organization (SISO)

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1730.1™-2013

IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO)

Sponsor

Simulation Interoperability Standards Organization (SISO)
of the
IEEE Computer Society

Approved 23 August 2013

IEEE-SA Standards Board

Abstract: A recommended practice for applying the Distributed Simulation Engineering and Execution Process (DSEEP) to the development and execution of distributed simulation environments that include more than one distributed simulation architecture is described. The distributed simulation architectures to which the recommended practice applies include Distributed Interactive Simulation (DIS), High Level Architecture (HLA), and Test and Training Enabling Architecture (TENA). The DSEEP Multi-Architecture Overlay (DMAO) identifies and describes multi-architecture issues and provides recommended actions for simulation environment developers faced with those issues. The DMAO also augments the DSEEP lists of inputs, recommended tasks, and outcomes with additional inputs, recommended tasks, and outcomes that apply to multi-architecture simulation environments. This document is an overlay to the DSEEP, which is a separate recommended practice.

Keywords: DIS, Distributed Interactive Simulation, distributed simulation, Distributed Simulation Engineering and Execution Process, DMAO, DSEEP, High Level Architecture, HLA, IEEE 1730™, IEEE 1730.1™, multi-architecture, systems engineering methodology, TENA, Test and Training Enabling Architecture

---

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

PDF: ISBN 978-0-7381-8659-7 STD98397
Print: ISBN 978-0-7381-8660-3 STDPD98397

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

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

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute 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 content 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. 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: PROCUREMENT OF 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 should be considered 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, or 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 his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide 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 revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

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 imply 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.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. 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 fee, 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. 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 ten years. When a document is more than ten 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 the IEEE-SA Website at http://ieeexplore.ieee.org/xpl/standards.jsp or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

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 http://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.

Participants

At the time this IEEE recommended practice was completed, the DSEEP Multi-Architecture Overlay (DMAO) Product Development Group (PDG) Working Group had the following membership:

Robert Lutz, *Chair* *
Roy Scrudder, *Vice Chair* *

Martin Adelantado
William Beebe
Jake Borah *
Dannie Cutts
Nico de Reus
Michael Egnor
Sarah Epps *
John Fay
Dan Gregory
Paul Gustavson *
Frank Hill

Jean-Louis Igarza
Kurt Lessman
Jennifer Lewis
Wayne Lindo *
Paul Lowe
William Luebke
Lance Marrou
James McCall
Bjorn Möller
Katherine L. Morse *

Eckehard Neugebauer
Mikel D. Petty *
Richard Reading *
John Rutledge
Steven Sheasby
Edward Shen
Robert Siegfried *
Marcy Stutzman
Cam Tran
Grant Tudor
Tom van den Berg *

* Denotes member of the PDG Drafting Group.

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

Tom Berg
Jake Borah
Juan Carreon
Keith Chow
Paul Croll
Sarah Epps
David Fuschi
Randall Groves
Frank Hill
Werner Hoelzl
Noriyuki Ikeuchi
Yuri Khersonsky
Murray Little

Greg Luri
Robert Lutz
Edward McCall
James McCall
Bjorn Möller
Katherine L. Morse
Charles Ngethe
Richard Reading
Robert Robinson
Christopher Rouget
Peter Ryan
Randy Saunders
Bartien Sayogo

Roy Scrudder
Steven Sheasby
Gil Shultz
Thomas Starai
Eugene Stoudenmire
Walter Struppler
Gerald Stueve
Marcy Stutzman
Cam Tran
Thomas Tullia
John Vergis
Jingxin Wang
Daidi Zhong

When the IEEE-SA Standards Board approved this recommended practice on 25 August 2013, it had the following membership:

John Kulick, *Chair*
David J. Law, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Peter Balma
Farooq Bari
Ted Burse
Wael William Diab
Stephen Dukes
Jean-Philippe Faure
Alexander Gelman
Mark Halpin
Gary Hoffman
Paul Houzé
Jim Hughes
Michael Janezic
Joseph L. Koepfinger*
Oleg Logvinov
Ron Petersen
Gary Robinson
Jon Walter Rosdahl
Adrian Stephens
Peter Sutherland
Yatin Trivedi
Phil Winston
Yu Yuan

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Michael Kipness
IEEE Standards Program Manager, Technical Program Development

Introduction

This introduction is not part of IEEE Std 1730.1-2013, IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO).

Modeling and simulation (M&S) has long been recognized as a critical technology for managing the complexity associated with modern systems. In many industries, M&S is a key enabler of many core systems engineering functions. For instance, early in the systems engineering process, relatively coarse constructive models are generally used to identify capability gaps, define systems requirements, and examine/compare potential system solutions. As preferred concepts are identified, higher-fidelity models are used to evaluate alternative system designs and to support initial system development activities. As design and development continues, very high fidelity models are used to support component-level design and development as well as developmental test. Finally, combinations of live, virtual, and constructive M&S assets are frequently used to support operational test and training requirements.

The advent of modern networking technology and the development of supporting protocols and architectures have led to widespread use of distributed simulation. The strategy behind distributed simulation is to use networks and support simulation services to link existing M&S assets into a single unified simulation environment. This approach provides several advantages as compared to development and maintenance of large monolithic stand-alone simulation systems. First, it allows each individual simulation application to be co-located with its resident subject matter expertise rather than having to develop and maintain a large stand-alone system in one location. In addition, it facilitates efficient use of past M&S investments because new, very powerful simulation environments can be quickly configured from existing M&S assets. Finally, it provides flexible mechanisms to integrate hardware and/or live assets into a unified simulation environment for test or training, and it is much more scalable than stand-alone systems. Examples of hardware and/or live assets are stimulators, live platforms, operational (command and control) systems, field instrumentation, and tracking devices.

There are also disadvantages of distributed simulation. Many of these issues are related to interoperability concerns. Interoperability refers to the ability of disparate simulation systems and supporting utilities (e.g., viewers, loggers) to interact at runtime in a coherent fashion. Many technical issues affect interoperability, such as consistency of time advancement mechanisms, compatibility of supported services, data format compatibility, and even semantic mismatches for runtime data elements.

Distributed simulation environments may involve the integration of hardware and/or live assets, and such integration creates additional issues. Examples of technical issues that are a consequence of integrating hardware and/or live assets include hard real-time execution; the mix of ground-truth (simulation) data with non-ground-truth (operational or live) data; and time, space, and position information (TSPI) updates.

The capabilities provided by today's distributed simulation architectures are designed to address such issues and allow coordinated runtime interaction among participating simulations. Three distributed simulation architectures are in common use in the M&S community and will be explicitly mentioned in this recommended practice, although these guidelines are not limited to those architectures. The widely used architectures are Distributed Interactive Simulation (DIS) as defined by the IEEE 1278™ family of standards, High Level Architecture (HLA) as defined by the IEEE 1516™ family of standards, and the Test and Training Enabling Architecture (TENA) used on test and training ranges. Associated with each of these simulation architectures is a systems engineering process for developing distributed simulation systems using that architecture. These processes, while individually effective, have been developed separately with the technical features, supporting facilities, and user community requirements of a specific architecture in mind and consequently are different from each other in both large and small ways.

In some situations, sponsor requirements may necessitate the selection of simulations, interfaces to live systems, and data collectors whose external interfaces are aligned with more than one simulation architecture. These situations lead to what is known as a multi-architecture simulation environment. When more than one simulation architecture is used in the same simulation environment, interoperability problems are compounded by the architectural differences. For such simulation environments to operate properly, reconciling middleware incompatibilities, dissimilar metamodels for data exchange, and differences in the nature of the services that are provided by the architectures is necessary. Developers have devised many different workarounds for these types of interoperability problems over the years. One possible solution is to choose a single architecture for the simulation environment and require all participants to modify the native interfaces of their applications to conform to it. While this solution is relatively straightforward and easy to test, it is usually impractical (particularly in large applications) because of the high cost and schedule penalties incurred. Another approach is the use of gateways, which are independent software applications that translate between the protocol used by one simulation architecture to that of a different simulation architecture (see Figure A). While effective, gateways represent another potential source of error (or failure) within the simulation environment, can introduce undesirable latencies into the system, and add to the complexity of simulation environment testing. In addition, many gateways are legacy point solutions that provide support only for a very limited number of services and only for very specific versions of the supported simulation architectures. Thus, it may be difficult to find a suitable gateway that fully supports the needs of a given application. For the relatively small number of general-purpose gateways that are configurable, the effort required to perform the configuration function can be significant and can result in excessive consumption of project resources.

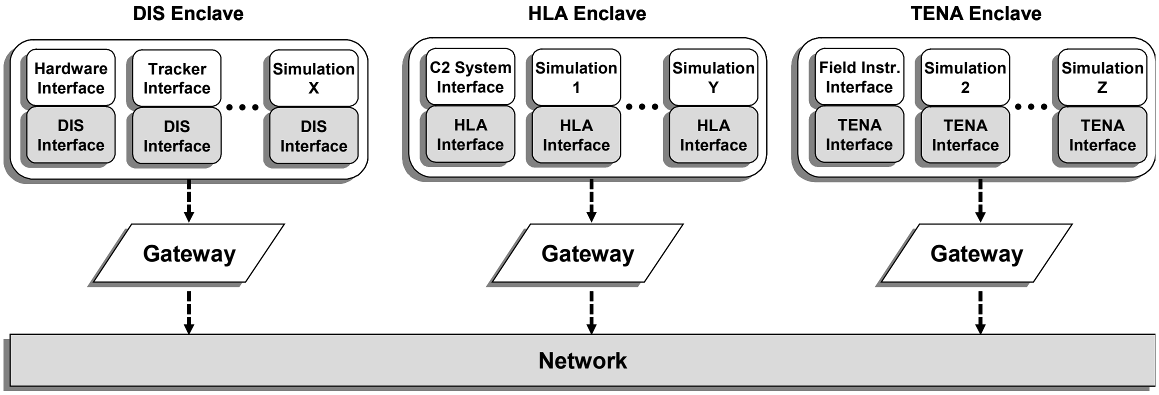

Figure A—Gateway configuration

The use of middleware is a similar approach but provides the translation services in software directly coupled to the simulation instead of an independent application[a] (see Figure B). While middleware approaches are also effective, they introduce many of the same technical issues that are associated with gateways (e.g., source of error, possible latency penalties). In general, all of these “solutions” have limitations and cost implications that increase technical, cost, and schedule risk for multi-architecture developments.

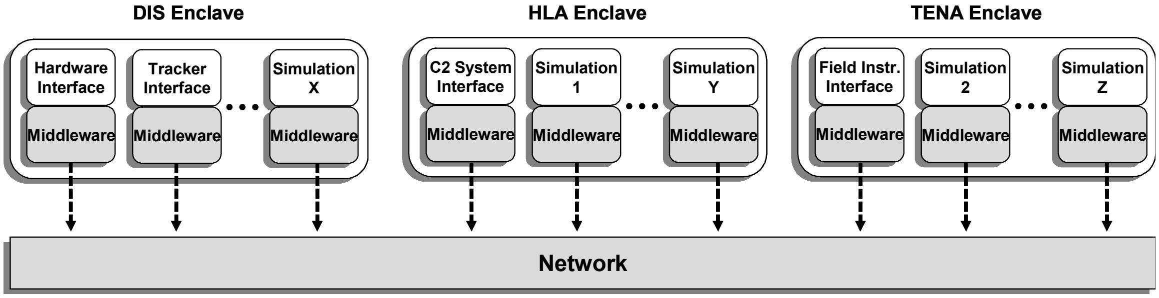

Figure B—Middleware configuration

---

[a] Note that this use of the term “middleware” is different in some user communities, who may use this term to refer to the infrastructure elements that provide distributed simulation services [e.g., the HLA runtime infrastructure (RTI)].

Many technical issues arise when multi-architecture simulation environments are being developed and executed. These issues tend to increase program costs and can increase technical risk and impact schedules if not resolved adequately. One widely reported and particularly vexing issue concerns the situation where users from different architecture communities are brought together to develop a single multi-architecture distributed simulation environment, but the differences in the development processes native to each architecture lead to misunderstandings, misinterpretations, and general confusion among team members. This situation impacts risk from many different perspectives and creates a persistent barrier to effective collaboration. An effective solution to this problem is to establish a common systems engineering process for the development and execution of multi-architecture simulation environments.

Many process standards already exist in the systems and software engineering communities. Rather than develop an entirely new process standard for multi-architecture simulation environments, leveraging and extending an existing standard seems to be much more efficient and to avoid the creation of potentially competing and conflicting products. While the principles that underlie existing systems and software standards (e.g., EIA-632, ISO/IEC 15288[b]) are certainly applicable, direct reuse of any process standard outside of the M&S domain would require a significant degree of tailoring. A more effective choice would be to select an existing distributed simulation engineering process standard, preferably one that is independent of any individual simulation architecture. The Distributed Simulation Engineering and Execution Process (DSEEP) document, IEEE Std 1730™, fits that requirement and provides a suitable choice for the desired multi-architecture process framework.[c]

The DSEEP is a generalized systems engineering process for building and executing distributed simulation environments, independent of the underlying simulation architecture. Intended as a high-level framework for simulation environment construction and execution, the DSEEP is the successor of architecture-dependent engineering processes, e.g., concerning HLA (IEEE Std 1516.3™ [B47]) or DIS (IEEE Std 1278.3™ [B42]).[d]

The DSEEP represents a tailoring of best practices in the systems and software engineering communities to the M&S domain and, in particular, to the development and execution of distributed simulation environments. The DSEEP is simulation architecture neutral, but it does contain annexes that map this architecture-neutral view to DIS, HLA, and TENA terminology. A top-level view of the DSEEP is provided in Figure C.

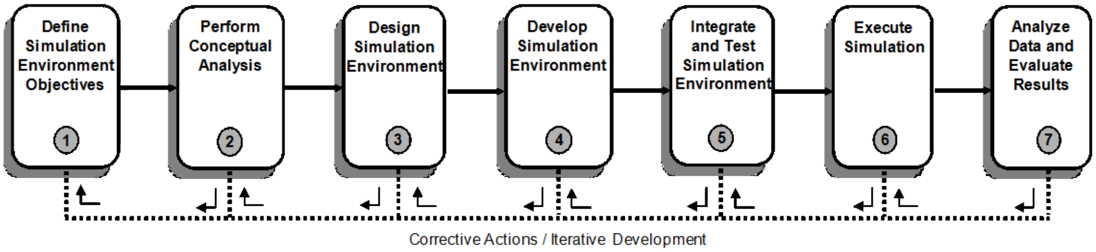

Figure C—DSEEP top-level process flow view

A short description of each of these seven major steps follows:

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 is to be accomplished to achieve those objectives.

---

[b] ANSI/EIA-632-1999, An aggregation of end products and enabling products to achieve a given purpose, and ISO/IEC 15288-2008, Systems and software engineering — System life cycle processes.

[c] Information on references can be found in Clause 2.

[d] The numbers in brackets correspond to the numbers in the bibliography in Annex A.

Step 2: Perform conceptual analysis. The development 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; design activities for member application modifications and/or new member applications are performed; required functionalities are allocated to the member applications; and a plan is developed for 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. All necessary integration activities are performed, and testing is conducted to verify that interoperability requirements are being met.

Step 6: Execute simulation. The simulation environment 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.

In the DSEEP document, each of these seven steps is further decomposed into a set of interrelated lower-level activities. Each activity is characterized according to a set of required activity inputs, one or more activity outcomes, and a list of recommended specific tasks. Although these activity descriptions are identified in a logical sequence, the DSEEP emphasizes that iteration and concurrency are to be expected, not only across activities within a step but across steps as well.

Although the DSEEP provides the guidance required to build and execute a distributed simulation environment, the implicit assumption within the DSEEP is that only a single simulation architecture is being used. The only reference to multi-architecture development in the DSEEP is provided in the following paragraph from DSEEP Activity 3.2 (design simulation environment):

“In some large simulation environments, it is sometimes necessary to mix several simulation architectures. This poses special challenges to the simulation environment design, as sophisticated mechanisms are sometimes needed to reconcile disparities in the architecture interfaces. For instance, gateways or bridges to adjudicate between different on-the-wire protocols are generally a required element in the overall design, as well as mechanisms to address differences in SDEMs. Such mechanisms are normally formalized as part of the member application agreements, which are discussed in Step 4.” (IEEE Std 1730)[e]

Clearly, additional guidance is necessary to support the development of multi-architecture simulation environments. However, the major steps and activities defined in the DSEEP are generally applicable to either single- or multi-architecture development. Thus, the DSEEP provides a viable *framework* for the development of the desired process, but it should be augmented with additional tasks as necessary to address the issues that are unique to (or at least exacerbated by) multi-architecture development. Such augmenting documentation is often referred to as an *overlay*. This document, IEEE Std 1730.1, constitutes such an overlay to the DSEEP, i.e., the DSEEP Multi-Architecture Overlay (DMAO).

In summary, the strategy implemented in this DMAO is to augment the major DSEEP steps and activities with the additional tasks needed to address the issues that are unique to (or at least exacerbated by) multi-architecture development. These tasks collectively define a “how to” guide for developing and executing multi-architecture simulation environments, based on recognized best practices.

---

[e] Throughout the document, text quoted directly from the DSEEP (IEEE Std 1730-2010) is both enclosed in quotation marks and highlighted with gray shading.

Contents

IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO)

1. Overview

A recommended practice for applying the Distributed Simulation Engineering and Execution Process (DSEEP) to the development and execution of distributed simulation environments that include more than one distributed simulation architecture is described. Distributed simulation architectures covered include Distributed Interactive Simulation (DIS), High Level Architecture (HLA), and Test and Training Enabling Architecture (TENA). This document identifies and describes multi-architecture issues; provides recommended actions for simulation environment developers faced with those issues; and augments the DSEEP lists of inputs, recommended tasks, and outcomes.

The DSEEP Multi-Architecture Overlay (DMAO) (IEEE Std 1730.1) is intended as a companion guide to the DSEEP (IEEE Std 1730™-2010).[1] The simulation environment user/developer should assume that the guidance provided by the DSEEP is applicable to both single- and multi-architecture developments. The DMAO provides the additional guidance needed to address the special concerns of the multi-architecture user/developer. In other words, the DMAO does not replace the DSEEP for multi-architecture simulation environment development, it augments the DSEEP.

---

[1] Information on references can be found in Clause 2.

1.1 Scope

This document defines the issues that are either unique to or exacerbated by the use of multiple simulation architectures in the same simulation environment, along with recommended actions for properly addressing these issues. The DSEEP (IEEE Std 1730) provides the overarching process framework. The alignment of DSEEP activities with those additional tasks necessary to address the multi-architecture concerns collectively define a "how to" guide for developing and executing multi-architecture simulation environments, based on industry best practices.

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 1730-2010, IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP).[2, 3]

3. Definitions, abbreviations, and acronyms

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.[4]

3.1 Definitions

"**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." (IEEE Std 1730)[5]

constructive simulation: Models and simulations that involve simulated people operating simulated systems. Real people stimulate (make inputs) to such simulations but are not involved in determining the outcomes.

issue: A concern, such as a situation within a development process or a technical element of an architecture, from which obstacles to achieving the objectives of the simulation environment may arise.

live simulation: A simulation involving real people operating real systems.

---

[2] IEEE publications are available from The Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[3] This IEEE document is a trademark of The Institute of Electrical and Electronics Engineers, Inc.

[4] *IEEE Standards Dictionary Online* subscription is available at http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

[5] Throughout the document, text quoted directly from the DSEEP (IEEE Std 1730-2010) is both enclosed in quotation marks and highlighted with gray shading.