RDECOM Modeling Architecture for Technology, Research and EXperimentation (MATREX) Support to Test & Evaluation M&S

Chris Gaughan, Chris Metevier, Lana McGlynn (RDECOM STTC); Michael Kierzewski (RDECOM ECBC); Jimmie Smith, John Diem (ATEC OTC)

MATREX Overview

The Modeling Architecture for Technology Research and EXperimentation (MATREX) provides a unifying distributed Modeling and Simulation (M&S) architecture and supporting tools and resources that ease the integration and use of multi-resolution live, virtual, and constructive (LVC) applications. It enables full spectrum analysis of system designs and operational concepts while reducing risk and acquisition timelines. To provide this capability across the Army and the other Services, the Assistant Secretary of the Army for Acquisition, Logistics, and Technology (ASA(ALT)) and the Research, Development & Engineering Command (RDECOM) has funded development of the MATREX program. The MATREX environment, including the toolset and resources, is available as Government-Furnished Software (GFS) with support and training available.

M&S customers who require the use of distributed simulation typically do not have a long life cycle for an experiment, analysis initiative or simulation-based event. To reduce cost, they need to use a well-established simulation architecture and robust models that are easy to integrate with other distributed simulations. This short lead time for system design, development, integration and execution forces the system definition and design to happen very quickly. This is where the various facets that compose MATREX can and do help.

For instance, one important aspect of MATREX is its capability to leverage many M&S resources developed at RDECOM laboratories, centers and activities. The world-class RDECOM team of M&S experts has utilized MATREX to identify issues, exercise requirements, align development efforts and conduct experiments that involve system of systems (SoS) technologies beyond the expertise of any one part of RDECOM. Figure 1 highlights the M&S contributions of each RDECOM component.

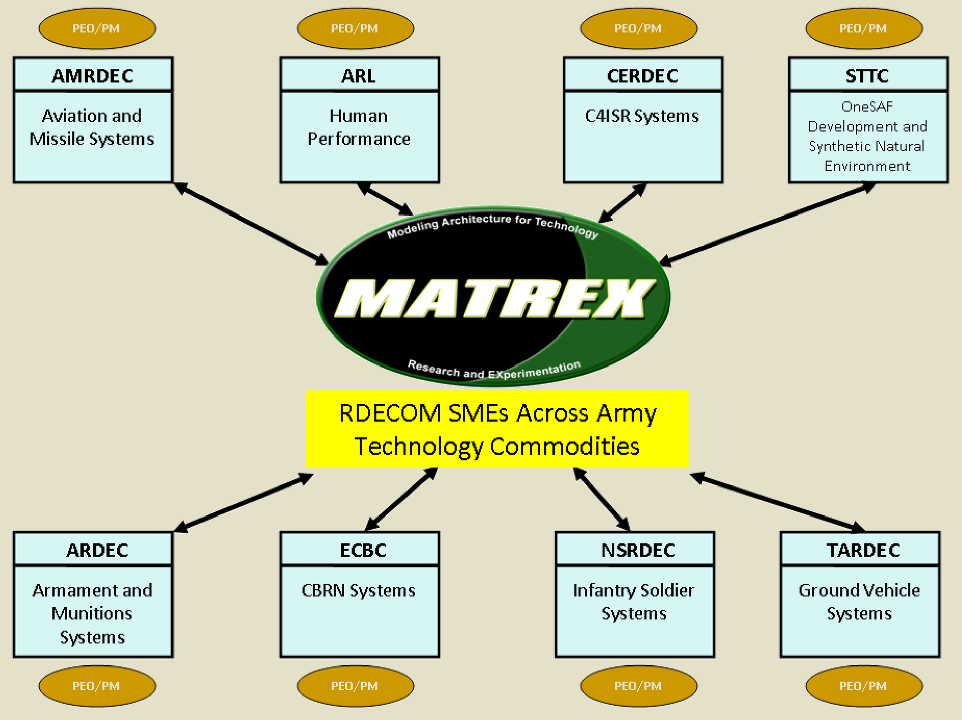

Figure 1 Contribution of the RDECOM Laboratories and Centers

Additionally, the MATREX environment is used for a variety of purposes. These numerous and often generic uses of MATREX offer a difficult systems engineering challenge in linking system requirements to detailed system design and technical dependencies. To facilitate customization for a given user's goals, the MATREX systems design team has developed a highly flexible and configurable system design that will allow it to meet a wide breadth of user requirements.

To determine how your organization can best use the capabilities of the MATREX program to meet your analysis needs it is useful to begin with an understanding of the system design approach and the supporting toolset and resources.

The MATREX System Design Approach

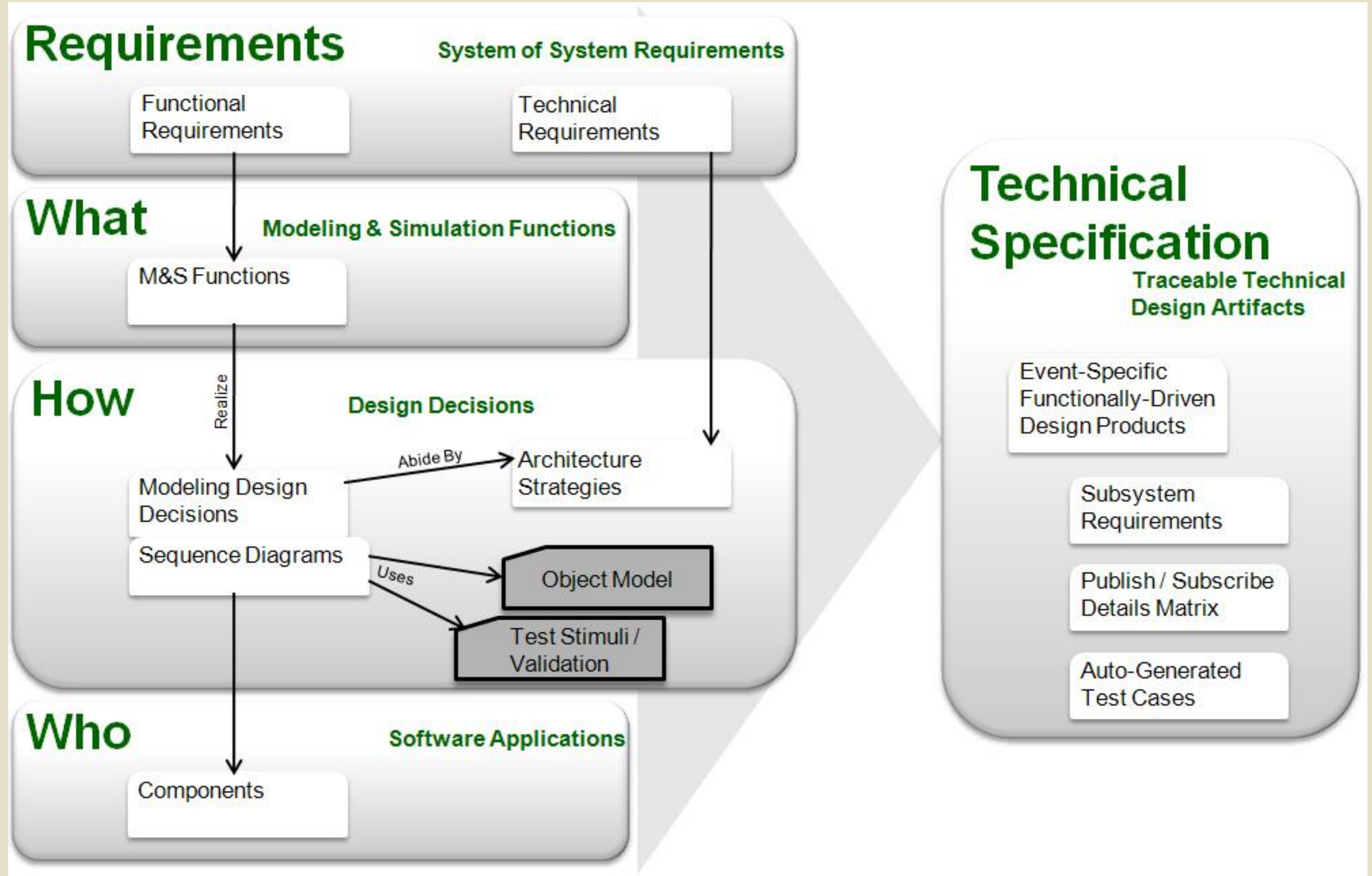

Figure 2 System Design Information Architecture

The MATREX suite of models, tools and architecture allow for many different possible configurations of the system to achieve the user's functional requirements. The philosophy is to work with the users to develop a System Design Description (SDD) that meets their exercise requirements, data decomposition requirements, system architecture guidelines, scenario, configuration choices and model selection. The SDD includes the mapping between the data to be collected during the exercise, the initial exercise goals and the explanation of what the data means.

The MATREX environment consists of models of various fidelities and resolutions. These models can be combined within an architecture that is designed to scale, so that, when the model permits it, multiple instances of each model can operate in parallel within the system. This added level of complexity allows more design possibilities (and challenges) than most simulation architectures. Simulation requirements such as "fair fight" issues, scalability concerns and data element analysis force the design to have additional architectural strategies that must uniformly be followed. These architectural strategies are captured and enforced with the SDD.

The MATREX program has developed the SDD to capture the system design at a functional level and subsequently link the functional design to the technical design. This allows the functional requirements to be linked to system design and allocated to specific models. The low level requirements, object model and test cases can be auto-generated based on model allocation to functions. The SDD is data-driven, easing information maintenance duties by linking the system engineering products and simplifying the editing of the system design. The SDD also allows for auto-generating low level specifications, certain DoDAF views and test cases.

The MATREX SDD exists within the MATREX Integrated Development Environment (IDE). The MATREX IDE is a content management system hosted on a secure server that provides various views into the MATREX system design. Since the SDD contains not only the object model information, but also the semantics of the data exchanges, test threads can be generated from the SDD and traced back to system requirements.

In addition, the test generation process uses a transport abstraction layer to allow these tests to be translated into various protocols for distributed simulation, e.g. High Level Architecture (HLA), Test and Training Enabling Architecture (TENA), etc., ensuring portability of the tests along with the models. Therefore, the tests evolve and migrate to new transport protocols with the SDD.

Figure 2 shows the four core building blocks MATREX uses to describe what the system does and how it accomplishes. They are: (1) Requirements, (2) What, (3) How and (4) Who.

In the "Requirements" block, the SDD contains System and Systems of Systems Requirements that depict two types of requirements: functional requirements which pertain to the model, and technical requirements which pertain to the execution architecture, such as, "The execution middleware will be HLA".

The "What" block contains the modeling and simulation functions that link to the functional requirements.

The "How" block contains the design decisions that describe how the functions are realized in the modeling design decisions (MDD). The architecture strategies which point back to technical requirements also must abide by the MDD. This enforces that the architectural decisions should dictate what models and capabilities are used instead of the model dictating what capability is available. The sequence diagrams for these design decisions can be realized are products which are then used by the object model and test stimuli/validation.

The "Who" block doesn't specify one model for one function but rather all of the models that have been evaluated, given the MDD, that fulfill that functionality. This allows for flexibility within events to use models based on fidelity or resolution requirements as well as political or monetary concerns, which unfortunately can dictate the architecture of an event despite gaps in capability.

All of these elements are combined to make up the MATREX System Requirements Specification (SRS), System Design Description (SDD) and System/Subsystem Specification (SSS).

MATREX Supporting Toolset and Resources

The MATREX program strives to create a system and tool suite that will ultimately facilitate a quick, automated system design to match the unique situation for systems engineering. The important elements of the MATREX system were kept as generic as possible to allow for various object models, middleware protocols such as HLA and TENA, design decisions and architectural strategies. The following is a list of the tools and services available to MATREX customers.

MATREX RTI and FOM

HLA is one of the simulation protocols available for executing a MATREX federation. HLA is implemented by using a Run-Time Infrastructure (RTI) and a Federation Object Model (FOM). MATREX has an RTI that is based on RTI-NG v9.0 and our RTI is provided as GFS. The MATREX FOM is co-managed with PEO-Integration and provides a basis for commonality in the PEO-Integration M&S community, RDECOM, TRADOC, ATEC, and beyond. There are many users across all Army commands that are using the FOM which increases interoperability across those organizations saving times and money across the M&S community.

ProtoCore

MATREX offers an Application Programmer's Interface (API) that abstracts away the middleware protocols in order to allow the developers of models to focus on model behavior vice rewriting code to align with specific distributed simulation protocols. This API, known as ProtoCore, reduces reliance on middleware services, data management techniques and architectural design patterns. In turn, it allows a simulation event to be middleware agnostic as the automated tool suite helps set up and configure models based on the SDD. Figure 3 shows ProtoCore.

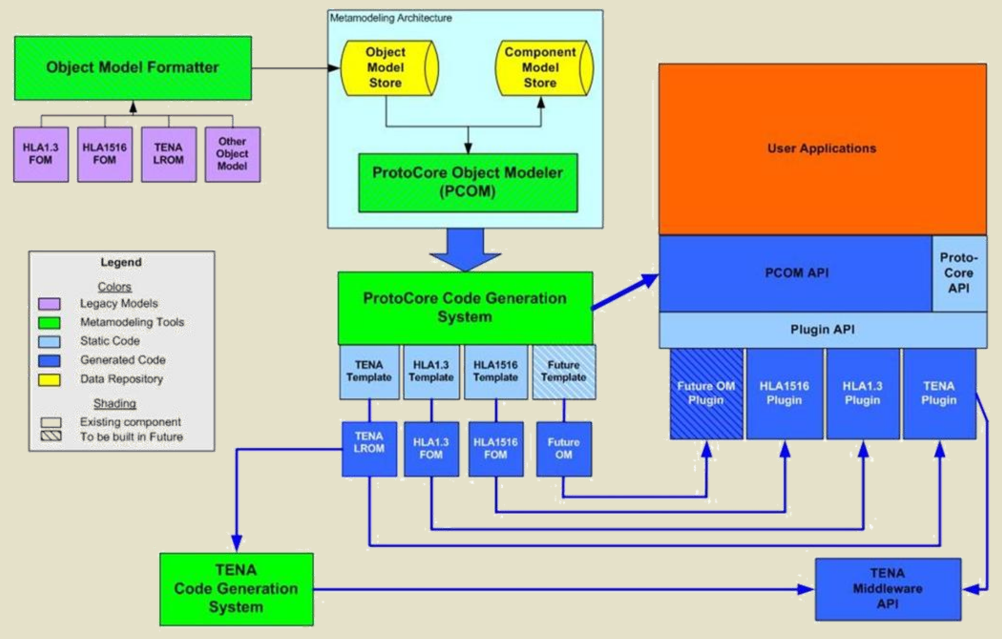

Figure 3: MATREX ProtoCore Middleware Independence Capability

Advanced Testing Capability

MATREX also offers a testing tool that generates test applications based on the design and over-the-wire communication distributed simulation requirements. The Advanced Testing Capability (ATC) allows for testing over the middleware layer based on a predefined sequence diagram including validation values. The ATC's primary purpose for MATREX integration testing is its ability to perform meaningful and repeatable "black box" level testing on any component being integrated into the MATREX environment. This allows a simulation to test interactions with other simulations without needing to run them in turn, reducing integration costs.

Battle Command Management Services

The MATREX offers a suite of models called the Battle Command Management Services (BCMS). The BCMS manages the Common Operating Picture (COP) and Local Operating Picture (LOP) for the force structure according to the information flow. It also provides interfaces for integration of external services including communication effects, fusion, human performance modeling and Command & Control devices. BCMS converts data from battlefield simulations into reported/detected information, models the reporting of that information from platform to platforms via defined reporting hierarchies, and maintains information about known entities (friendly & enemy) on behalf of the simulation(s) modeling the battlefield entities.

Configuration & System Administration Tool

MATREX uses the Military Scenario Development Environment (MSDE) to generate data in a Military Scenario Definition Language (MSDL) format. To accommodate the full set of data needed to initialize a distributed simulation, MATREX has developed a tool called the Configuration and System Administration Tool (CSAT). CSAT ingests a scenario produced in MSDL format and allows a user to supplement the scenario data with simulation configuration data. The output of CSAT is then used to initialize the models within a simulation.

Distributed Virtual Laboratory

The MATREX DVL provides a closed classified distributed networking capability to all RDECOM laboratories and support centers. The DVL offers the capability to conduct experiments and make demonstrations of MATREX studies, concepts and capabilities. The MATREX DVL also peers with the Cross Command Collaboration Effort (3CE) Network allowing the MATREX community access to the TRADOC Battle Lab Simulation Collaboration Environment (BLCSE) network, ATEC Test Integration network (ATIN) and the PEO-Integration network. Services provided by the MATREX DVL include: systems engineering and architecture development; design and development; integration and test; performance analysis; operations and technical support; network security infrastructure;.

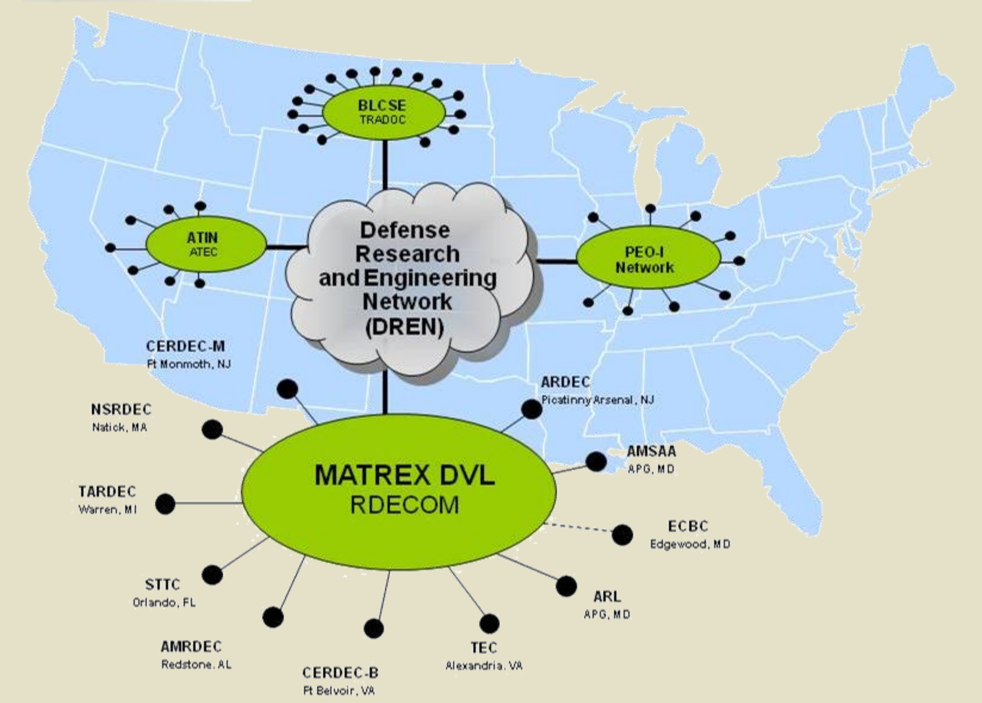

Figure 4: MATREX Distributed Virtual Laboratory

Integrated Development Environment

The Integrated Development Environment (IDE) is the content management system for the MATREX program. This system facilitates distributed systems engineering, configuration management and sharing of key information via the Internet.

Example Usage of MATREX

The US Army Operational Test Command (OTC) provides an example usage of the MATREX environment. MATREX and OTC developed a beneficial partnership through sharing technologies and capabilities. In particular, MATREX and OTC leveraged the feasibility of Cross Command solutions to modeling, simulation, testing, and validation of legacy, current and future systems and architectures. OTC met its near-term mission requirements for integration through a series of integration events utilizing an HLA approach in the form of the MATREX FOM and RTI enabled by porting OTC simulations to ProtoCore. These events, occurring regularly over the number of years, have systematically increased in level of complexity and scope (35 enterprise members), culminating in a seamless integration of live, virtual and constructive entities made possible, in part, by a robust MATREX FOM and RTI.

The OTC Analytic Simulation and Instrumentation Suite (OASIS) integrates test technologies and weaves them into the live test environment by providing a family of integrated systems to support operational tests. It provides an acquisition strategy to adapt, buy or create the common components for a family of integrated, interoperable enterprise tools to support test technology "centers of gravity" which includes: live, virtual, constructive (LVC) environments of integrated warfighting systems; tactical systems including system under test (SUT); test control and networks which is part of the environment; and data collection, reduction and analysis (DCRA), which is critical in order to assess the SUT performance. There have been enumerations added to the FOM in order to better model/represent live entities on the battlefield – a benefit to all the members of this collaboration.

Perhaps the greatest example of this collaboration was demonstrated when components of OASIS integration effort particularly the Extensible C4I Instrumentation System (ExCIS) fire support simulation were integrated with simulations and simulation support tools from training and R&D applications to create a unique FCS Spin-out 1 (SO1) federation that satisfied the requirements of the four very different event stakeholders trainers, doctrine developers, material developers, and operational testers, which own phases of the SO1 event window. The SO1 simulation team was able to achieve, at least in one small but very visible way, the promise of shared common components and standards because each of the tool kit providers, regardless of domain, had worked during the previous year to integrate the MATREX FOM in support of other efforts.

Conclusion

To reiterate, the goal of the MATREX program is to develop a composable M&S environment wherein a collection of multi-fidelity models, simulations, tools and resources can be integrated and mapped to an established architecture for conducting analysis, experimentation and technology trade-offs for RDECOM and others. MATREX is both a concept and an environment, not a model. It is architected to integrate existing M&S into a robust representation of the battlespace, ultimately seeking to combine the capabilities of the Army's highest fidelity digital terrain, dynamic environmental effects and physics-based M&S. The unifying architecture of MATREX allows integrated models to pass data among themselves and share a common synthetic battlespace. The MATREX Team has developed a cross-cutting set of tools that facilitates interoperability and ease of use.