13TH ICCRTS

Thirteenth International Command and Control Research and Technology Symposium

C2 for Complex Endeavors

June 17 – 19, 2008

The Meydenbauer Center

Bellevue, WA

Topic: Topic 6: C2 Assessment Tools and Metrics

Title: Testing C2 Interoperability – Advancements in Testing of the MIP Baseline 3 Solution

Authors:

Name: Nico Bau
Michael Gerz
Michael Glauer

Organization: FGAN FKIE

Address: Neuenahrer Straße 20
53343 Wachtberg-Werthhoven, Germany

Phone: +49 228 9435 414

Fax: +49 228 9435 685

E-Mail: {bau|gerz|glauer}@fgan.de

Testing C2 Interoperability — Advancements in Testing of the MIP Baseline 3 Solution —

Nico Bau, Michael Gerz, Michael Glauer

Abstract

Semantic interoperability of command and control information systems (C2ISs) is critical to combined and joint missions. Thus, in the Multilateral Interoperability Programme (MIP), 26 nations and NATO define a common interface for distributed, heterogeneous C2ISs.

In order to support testing, MIP defines a set of test specifications as well as a schedule with fixed test periods. The tests are conducted in bi- or multilateral sessions during which the participating C2ISs switch their roles of being evaluated and evaluating the partner(s). Since interoperability between two systems does not necessarily mean conformance to the specification, these sessions are repeated with multiple partners. Thus, test coordination and execution tends to be time consuming and error-prone.

The new MIP Test Reference System (MTRS) provides an infrastructure to perform tests at any time. All test documents are formalized in a way that allows for automatic test execution and describes a test scenario unambiguously. Moreover, the MTRS provides means for efficient test evaluation and for identifying potential errors in the implementation.

In this paper, we will describe the MIP test procedures and their implications on interoperability and conformance. Besides a general overview of the MIP tests and the MTRS, we will provide some insight on the lessons learned while developing and using the MTRS.

Keywords: Multilateral Interoperability Programme, MIP, Joint Command Control and Consultation Information Exchange Data Model, JC3IEDM, Data Exchange Mechanism, DEM, Command and Control Information Systems, C2IS, Conformance Testing, MIP Test Reference System, MTRS

1 Introduction

Semantic interoperability of *command and control information systems* (C2ISs) is critical to combined and joint missions. Thus, in the *Multilateral Interoperability Programme* (MIP), 26 nations and NATO define a common interface for distributed, heterogeneous C2ISs.

The intent of MIP is "to achieve international interoperability of C2IS at all levels from corps to battalion, or lowest appropriate level, in order to support multinational (including NATO), combined and joint operations and the advancement of digitization in the international arena."

(MIP, 2006). For this purpose, MIP provides consensus-based, system-independent technical specifications to achieve automated information exchange. The MIP core features are the Joint Command, Control, and Consultation Information Exchange Data Model (JC3IEDM) and the Data Exchange Mechanism (DEM).

The MIP specifications, called *baselines*, are developed in an incremental manner based upon overlapping *blocks*. Each block comprises a development phase, in which the MIP members analyze operational requirements, develop concepts, make feasibility studies, define the MIP solution, implement it in their national systems, and finally perform interoperability tests. By the end of this phase, the baseline is finished and the baseline and the MIP-compliant systems enter an in-service period. For the forthcoming MIP baseline 3, initial draft specifications have been available on the public web site since December 2006 (MIP, 2007a) and interoperability tests have started in September 2007.

As extensive and efficient testing of the national MIP implementations is critical to ensure interoperability even in exceptional situations, a large set of tests and an extensive schedule with fixed test periods are defined. These tests cover the data exchange between two MIP gateways, the common data model, the exchange between C2ISs, and a complex operational scenario.

In the bi- or multilateral test sessions, each participating C2IS switches its role of being evaluated and evaluating the respective partner(s). If the test criteria is met (i.e. the correct data is exchanged between the participants in all roles), the systems are considered to be interoperable in the tested area. These sessions have to be repeated with multiple partners, as interoperability between two systems does not necessarily mean that both are conforming to the specification. This approach makes test coordination and test execution time-consuming, in particular, as C2ISs are not designed to explicitly support testing, test preparation and evaluation are expensive.

The *Research Institute for Communication, Information Processing, and Ergonomics* (FKIE) develops the *MIP Test Reference System* (MTRS) to support testing at any time with no coordination overhead. All test documents are formalized in a way that allows for automatic test execution and describe a test scenario unambiguously. Moreover, the MTRS has several features for efficient test evaluation and for identifying potential errors in the implementation.

Guide through this Document. This paper is structured as follows: In section 2, we will give an overview of the technical solution of MIP Baseline 3 as defined by the current draft specifications. The knowledge about these core concepts is needed to understand the MIP testing concepts discussed in section 3. The MTRS is introduced in section 4. The practical application of the MTRS allowed us to gain experience with testing in an international context with many different systems. Statistics on the utilization of the MTRS and the lessons learned so far are described in section 5. Finally, in section 6, a brief summary and outlook is provided.

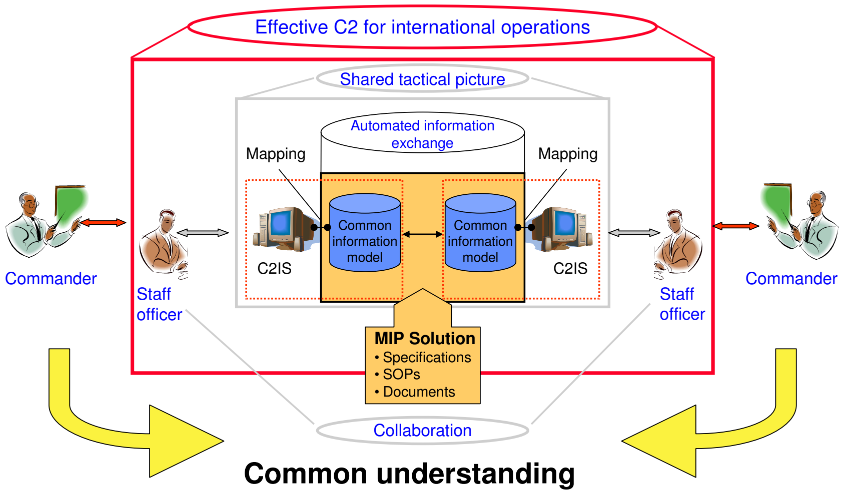

Figure 1: The MIP Solution (figure adopted from MIP briefing)

2 MIP Baseline 3

The MIP Baseline 3 Solution has been presented in detail at the 12th ICCRTS (Bau et al., 2007b). In the following, we give a short summary of the major technical aspects.

The MIP Solution is a set of specifications to support automatic information exchange between distributed and heterogeneous C2ISs. The MIP Solution and its integration into the C2 environment are shown in figure 1.

On the technical level, interoperability is achieved by two features:

- A common information exchange data model
- Information exchange protocols and procedures

The technical aspects of these solutions are explained in what follows.

2.1 JC3IEDM

A core feature of the MIP solution is the *Joint Command, Control, and Consultation Information Exchange Data Model* (JC3IEDM; see MIP, 2007c). It is developed in close cooperation with the *NATO Data Management Service Working Group (NDMSWG; former NATO Data Administration Group)*, which signed an agreement with MIP in February 2004. The JC3IEDM provides the basis for information exchange and specifies the semantics of militarily relevant objects, actions, etc., as well as the semantics of their relationships in an unambiguous way.

The JC3IEDM is an Entity-Relationship model in IDEF1X notation (IEEE, 1998). It defines all military information in terms of entities (e.g., UNIT, HARBOUR, LOCATION), optional or mandatory attributes, domains (e.g., codes with a finite set of predefined values), and relationships among entities.

Moreover, MIP defines a large set of business rules that specify the proper use of the ER model. For instance, some types of capabilities can only be assigned to specific types of equipment. In baseline 3, many of these business rules are available in a formal representation, which simplifies their implementation.

The most recent version of the JC3IEDM is 3.1b. It is the second bug fix release of the 3.1 series, which will serve as the basis for MIP baseline 3. In comparison to the C2IEDM 6.15e of MIP Baseline 2, the MIP data model has been extended significantly. Presently, it comprises 271 entities and 2005 attributes.

2.2 Data Exchange Mechanism

In addition to the data model, MIP also defines the *Data Exchange Mechanism* (DEM). For MIP baseline 3, the DEM was redesigned based on the lessons learned in the previous block. This has resulted in a much simplified technical implementation, easier administration and operation of MIP gateways, and enhanced reliability and robustness.

The DEM supports the partial replication of operational data depending on their affiliation to a particular *Operational Information Group* (*OIG*). This OIG concept separates the information space into operationally distinct groups, which can be delivered to different receivers. MIP defines six different OIG types: *Friendly and neutral (both organisational and non-organisational)*, *Enemy and unknown (both correlated and uncorrelated)*, *Globally significant*, and *Plans and orders*.

The DEM uses a publish-subscribe mechanism to distribute data, where each subscription refers to a particular OIG of a particular organization. Within each group, referential integrity and semantic completeness of information is required by the replication rules of the DEM. Furthermore, the DEM checks for the compliance to JC3IEDM business rules. For instance, if an organization is added to an OIG, the DEM takes care of transmitting the organization's hostility status as well.

Each MIP gateway announces itself by sending its *DEM Connection Information* (DCI) in a UDP broadcast message to all other MIP gateways on a MIP LAN (In case of WANs, DCIs can also be exchanged manually). Among others, the DCI contains the node id of a MIP gateway, its IP address and TCP port, as well as its supported MIP version. The latter is useful if two gateways support different MIP baselines and the more recent gateway is backward compatible. As a reply, all MIP gateways on the LAN will send their own connection information via unicast to the IP address provided in the newly received connection information.

When a node connects to another node, the latter will publish its list of available OIGs. If a node subscribes to an OIG, the provider sends operational data contained within that OIG to the subscribed node. The DEM protocol specifies two messages, SUBSCRIBE and UNSUBSCRIBE, which allow the receiver to change the subscription status for a list of OIGs. In addition, the data provider may indicate that an OIG is no longer available.

The DEM of MIP baseline 3 uses synchronization points to reduce the required bandwidth in case of connection loss. A synchronization point is assigned to each data message (each

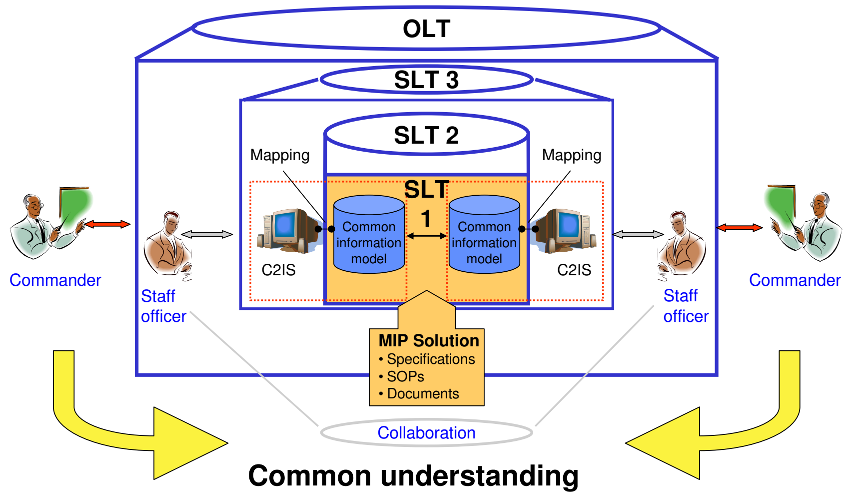

Figure 2: MIP System- and Operational-Level Testing

protocol data unit (PDU) contains data of only one OIG). If transmission fails (e.g., due to a network problem), the data receiver is able to re-subscribe to the OIG later. When sending its subscription message, the data receiver may provide the synchronization point of the last PDU it was able to process for that particular OIG. This synchronization point allows the sender to determine the delta it has to send in order to update the receiver instead of transmitting all data again.

3 MIP Interoperability Testing

The MIP community defines a multi-step testing strategy in order to ensure that their C2 information systems are interoperable. Testing takes places at three distinct levels (see MIP Test and Evaluation Master Plan, 2007d):

- *Implementation Level Tests* (ILT) are conducted under national responsibility.
- *System Level Tests* (SLT) demonstrate the timely end-to-end transfer of operational data between national C2ISs .
- *Operational Level Tests* (OLT) evaluate the MIP Solution, when deployed in the context of an operational scenario, and validate that the MIP Solution meets the operational objective.

The System Level Tests are further divided into three sub groups:

- SLT 1 focuses on the communication protocols (protocol level).
- SLT 2 focuses on the information exchange between JC3IEDM databases (database level).
- SLT 3 focuses on the information exchange between C2ISs (C2IS level).

Figure 2 illustrates how SLT 1 to 3 and the OLT fit into the global picture.

For each SLT test group, a distinct set of test cases (including comprehensive test data) is specified. In the following, the test groups are described in detail.

System Level Test 1. The objective of the SLT 1 test cases is to validate the protocol stack implementations. Accordingly, many test cases are derived from the state-transition table of the DEM protocols. They check whether the MIP gateway responds correctly according to the MIP specification if it receives a certain stimulus. SLT 1 also defines test cases that check the behavior of the MIP protocols in case of inopportune or invalid input. For instance, a system must be able to cope with a connection loss in any state.

In addition, several test cases are concerned with messages that violate the PDU grammar or that are semantically broken. For example, a system must be able to handle invalid special characters gracefully and reject subscriptions to unknown OIGs.

System Level Test 2. While SLT 1 increases faith in the publish-subscribe mechanism as such, SLT 2 is concerned with the replicated operational data. As mentioned above, the MIP DEM requires the replicated data to be referentially and semantically complete. Moreover, there are rules on what data records need to be replicated depending on the linkage to some OIG or to other data. Accordingly, most SLT 2 test cases check whether the right data is replicated. Technically, this is achieved by iteratively applying the following test steps:

1. DEM A (the system under test) imports some JC3IEDM data off-line.
2. DEM B (the test system) subscribes to a specific OIG of the system under test.
3. As a result of the subscription, DEM A sends data to DEM B.
4. DEM B checks whether the received data matches with the expected (subset of) data.

As part of the test specification, the MIP community has defined a large repository of test data in SQL INSERT statement format. Annotations are used to indicate that a given data record must – or must *not* – be available in the receiver's database after a specific test step. Based on the annotated SQL scripts, files for offline updates and SQL scripts that check the existence of database records after each test step can be generated automatically. Moreover, the FKIE has developed a tool that validates the SQL annotations for several consistency and completeness criteria. (see `https://trac.fkie.fgan.de/MTRS/browser/trunk/MIP/tools/CheckScriptGenerator/` for details.)

In addition to the tests described above, there are tests that check the behavior of MIP gateways in case of erroneous input. Test scenarios include referentially incomplete data, domain value violations, missing attributes in data records, etc. Since the JC3IEDM meta model is available in a formal representation (MIP Information Resource Dictionary), it was possible to generate most of these tests automatically.

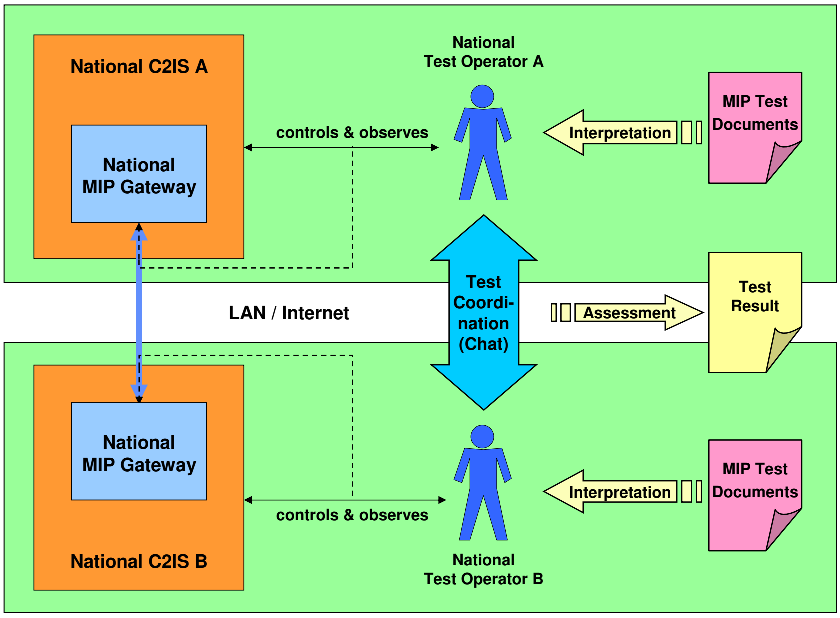

Figure 3: Bilateral Interoperability Tests

System Level Test 3. SLT 3 tests the semantic integrity of data between national C2ISs across the MIP gateways and confirms operational interoperability between C2ISs. It should be pointed out that the integration of the MIP Solution is a national issue. Nevertheless, it must be tested whether a C2IS maps user inputs correctly into the JC3IEDM and vice versa. In particular, this holds for objects represented as graphical symbols on screen (e.g., based on APP6a) and for their geometry.

Operational Level Test. The OLT confirms operational interoperability between C2IS systems under near-real operational conditions. It evaluates the reliability, functionality, and performance (application and system performance) of the MIP Solution under test. Accordingly, the test specification activities are mainly focused on developing a realistic operational scenario, which has to be performed in several independent test sessions with a varying number of participants as well as with different partners.

Test Procedure. The MIP community test its implementations during bilateral and multilateral interoperability test sessions, either via the Internet or co-located during fixed periods in Greding, Germany. A generic test configuration for bilateral interoperability tests is shown in figure 3.

In this scenario, two C2ISs A and B are controlled by their national test operators and their behavior is compared with the expected results specified in the MIP test documents. In some cases, the test operators have to disrupt the underlying MIP LAN in order to test the behavior of their implementations. During test execution, the test operators must synchronize their activities off-line, e.g., by a chat tool. At the end, they have to determine the final test verdict. All test cases are executed twice, with the national systems swapping their roles.

Unfortunately, this testing approach shows several limitations:

- The informal or semi-formal MIP test documents easily become subject to interpretation and the test results depend on the judgment of the test operators, as indicated in figure 3.
- Since there are no executable test scripts, all test cases have to be performed manually, following the instructions in the test documents.
- Testing has proven to be rather time-consuming, because testers must coordinate with each other.
- Errors remain undetected if the systems, which are tested with each other, are implemented in the same erroneous way. This problem is addressed by testing a C2IS against as many other C2ISs as possible (at least 3 to 5 depending on the system level).
- Since a lot of time is needed to set up a common test configuration (including filling/resetting the operational database), it is practically impossible to iteratively execute thorough regression tests.

These limitations have resulted in the development of the MIP Test Reference System, which is described in the next section.

4 The MIP Test Reference System

The FKIE develops a test system tailored to the specific needs of the MIP community that makes use of formal test cases and allows for the automated execution and evaluation of test cases.[1] The test system supports the full range of MIP system level tests[2]. It is offered free of charge to all systems participating in MIP and is used as the *MIP Test Reference System* for the official MIP SLTs.

Unlike the MIP interoperability tests, the MTRS does not check for the interoperability of two C2IS but aims at checking the conformance (i.e. functional behavior) of a *single* C2IS to the MIP specifications. While testing, the C2IS is considered as a *black box* that is triggered with some input and whose observed output is compared with the expected results.

The only common, standardized interface of the C2ISs under test is the MIP interface. Accordingly, the abilities of the MTRS to control and observe a C2IS are limited. If the PDUs sent and received via MIP do not allow for the complete execution and evaluation of a test case, the test operator has to be involved in a manual step. For instance, the MTRS may ask the operator to create and replicate some military symbols via the user interface of the C2IS.

[1] The project is funded by the Federal Office of the Bundeswehr for Information Management and Information Technology.

[2] The only exception are tests for the *Message Exchange Mechanism* (MEM). The MEM is an email-based alternative to the DEM, which is only used for a few NBC (Nuclear, Biological, Chemical) and system administration messages.

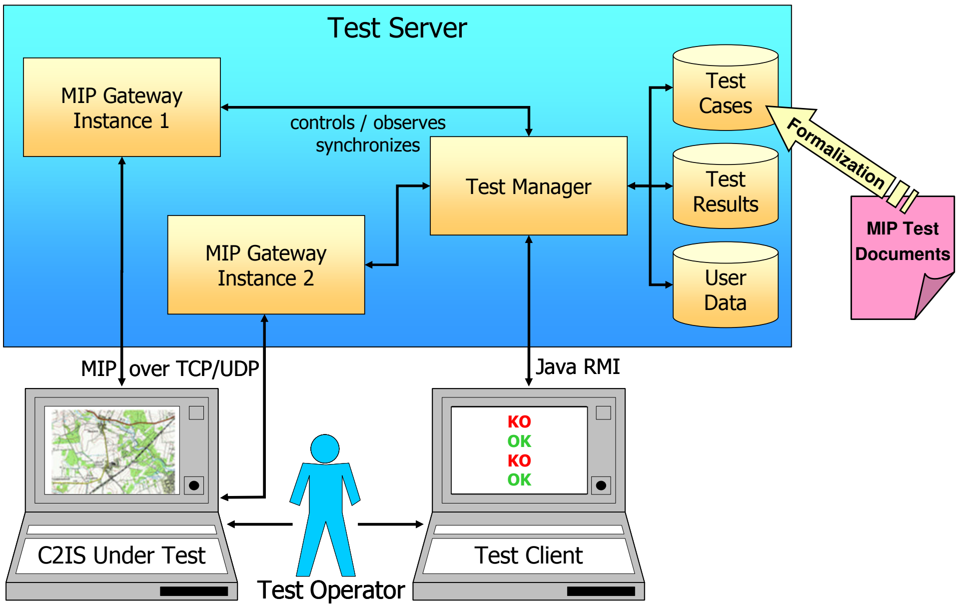

Figure 4: MTRS Architecture

In accordance to the SLT subcategories, testing of the MIP implementations must happen on different technical and logical layers (protocol layers, database layers, C2IS user interface). Likewise, test cases are specified on different levels of abstraction. They make use of MIP software components that are integrated into the test system itself and implement the lower layers. The MIP-specific modules have been modeled as closely to the MIP specifications as possible. By means of a light-weight component framework, varying test configurations can be set up. Moreover, the framework provides the test operator with information on the data flow between the components. The latter enables efficient error diagnosis and test evaluation. The test configuration is specified as part of each formal test case (see section 4.3).

4.1 MTRS Architecture

The MTRS is a client-server application, whose architecture is shown in figure 4. The national test operator interacts with the MIP test system via the *test client* (see section 4.2). Using a graphical user interface, the national test operator is able to run test cases and to analyze the test case logs. The MTRS also offers an application programming interface (API) to run tests in batch mode.

All tests are executed and evaluated on the *test server*. The *test manager* supports concurrent execution of test cases for different C2ISs. It is also responsible for setting up, controlling, observing, and synchronizing specific test components. In figure 4, two MIP gateway instances are set up. This test configuration is useful when testing the data forwarding capabilities of a C2IS. In this scenario the test manager sends operational data via MIP gateway 1 and checks for the reception of the same data at MIP gateway 2.

The MTRS test suite, the test results of each individual C2IS, and the user data is made persistent in the server database.

As mentioned above, the MIP interface is the only standardized interface provided by all national C2ISs. Therefore, at certain points, the test server sends action requests to the test client, which then asks the test operator to perform some action or to provide feedback on what information is displayed at the C2IS's user interface.

Using a common test server has several advantages:

- The integrity of test cases and test results is ensured by a central server repository – the test operators are not able to alter them.
- Since all test results are available in one place, cross-national test reports can be produced for the MIP test controllers at run time. In fact, the MTRS allows to export such reports in PDF format.
- The update procedure for the test system and for its test scripts is strongly simplified, because no distribution among the nations is needed. Test scripts can be updated on the fly. For that purpose, the server administrator connects to the MTRS server via a special administration tool and uploads the scripts.
- The national testers do not have to install and configure the test system locally but only need to download the MTRS client and a recent Java Runtime Environment. The server configuration and maintenance as well as the database backups are at the responsibility of a single organization.

Then again, a server-based system has much higher requirements regarding reliability and availability. Special care was taken to ensure that a broken test script or a faulty test component cannot tear down the complete server or make it inoperable. Moreover, all MIP components are designed in a way that they do not interfere if more than one C2IS is tested at the same time. On the technical level, this has been achieved by encapsulating all error-prone server components in a sandbox that catches and handles all exceptions. Furthermore, long-running test cases and user sessions with no activity are terminated based on watchdog timers. Since the MTRS is used of the Internet, all test data must be unclassified. However, this is not a problem, since all tests are publicly available anyway.

4.2 Test Client User Interface

The MTRS client interface is shown in figure 5. The GUI is split into three main areas: On the left, a tree view of the test suite with its test groups, test cases, and corresponding test runs is given. On the top right, meta information is provided for the currently selected tree node. A test case is, among others, characterized by its name, its version, and its purpose, and a set of keywords. Keywords (like *Unwritten*, *Extra*, etc.) can be used for filtering test cases in the tree view and in the national test reports. For a test run, the MTRS keeps track of, e.g., the status (running, terminated, aborted, etc.), the test verdict, start date and time, etc. The MTRS supports three possible test verdicts: *pass*, *fail*, and *inconclusive*. Test verdict *inconclusive* is

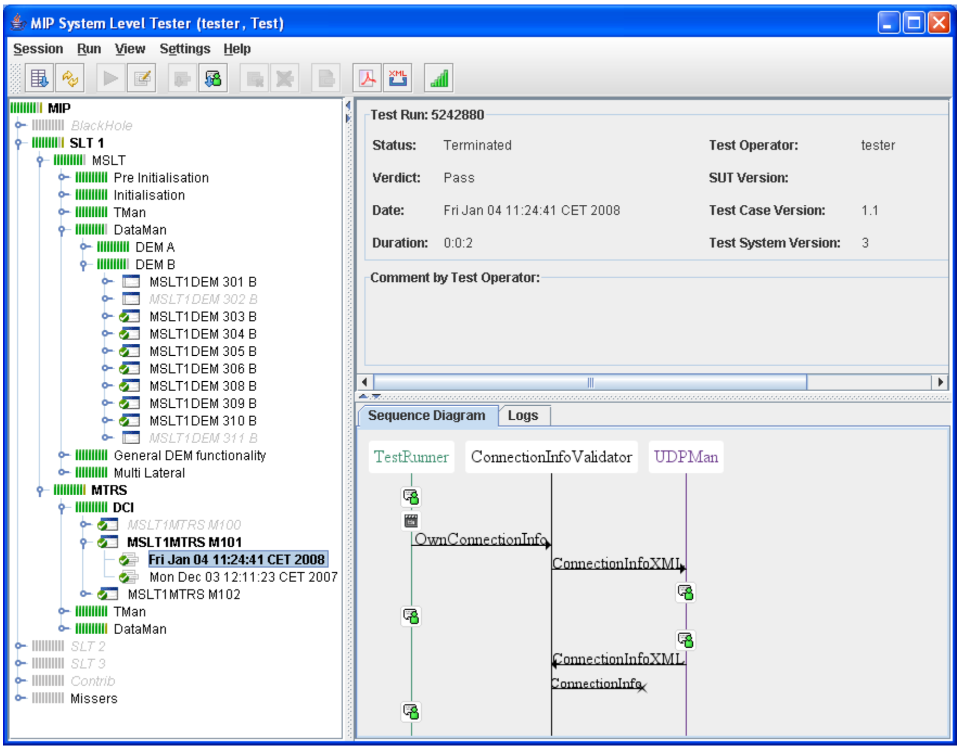

Figure 5: MTRS Client Screenshot

assigned if the C2IS does not meet the test objective but behaves correctly according to the MIP specifications. In most cases it indicates a user abort. The internal data flow between the MTRS MIP gateway components and various status messages are shown on the bottom right. Besides displaying the log information in a tabular view, the MTRS is also able to create sequence diagrams.

4.3 Example

The following extract of the MIP specification illustrates the development process of formalizing a test case. The *MIP Technical Interface Design Plan* (MTIDP; MIP, 2007b) specifies the DEM discovery process as described in section 2. It provides an XML schema as a message format for the DEM Connection Information. This DCI contains information about the MIP gateway. In particular, it includes a node id to identify the node, an IP address and a TCP port, on which the MIP gateway listens. The following business rules are quoted literally from the MTIDP 3.5, chapter 5.3.3:

- DEMs shall listen for UDP messages on port 13152.
- DEMs shall send their DCI using the 255.255.255.255 broadcast address when:
  - the DEM is first started
  - manually invoked by an operator

  - the DCI is updated

- When sending DCI using a broadcast address, the scope of the node the contained XML document will be set to ANNOUNCE.
- DEMs will respond to received DCI messages where the scope of the node is set to ANNOUNCE by sending their own DCI with the scope of the node set to REPLY.
- The reply will be sent using the unicast IP address from which the ANNOUNCE DCI was received. Nations deeming that automatically replying to a DCI constitutes a security risk may respond as a result of operator action or not at all. This is not recommended.

Several test cases were specified to test the correct implementation of these rules. In test case *MSLT1DEM M101 (DCI: Replying to an announced DEM Connection Information via unicast)* a simple scenario is tested: DEM B (MTRS) sends its own DCI to DEM A (the C2IS). DEM A replies to the DCI by sending its own DCI. In the test document, the following steps are described:

Step	Action	Expected Result
1.	DEM B sends its DCI with scope ANNOUNCE to the specified IP address of DEM A.	DEM A receives the DCI of DEM B.
2.	DEM A sends its DCI with scope REPLY to IP address in the received DCI of DEM B.	The test is passed if DEM B receives a DCI with scope REPLY.

As this test should be performed over the Internet, DEM B has to announce its DCI via unicast to DEM A. However, this is something that is not required by the specification, as the specification only requires the ability to send a DCI with scope ANNOUNCE to the specified broadcast address. Thus, only systems that have implemented an additional functionality can conduct this test with the role of DEM B.

This test, as simple as it looks, contains several ambiguities that are open to interpretation:

- What happens if DEM A sends some other data before sending its DCI?
- What happens if the information provided in the DCI is wrong (e.g., an unknown node id is given)?
- What happens if DEM A repeatedly sends its DCI with Scope ANNOUNCE?
- How long should DEM B wait for the reception of the DCI?

If these questions were answered differently by different test operators, this might result in different test verdicts for the same behavior during bilateral testing. Therefore, these ambiguities have to be resolved when formalizing the test case, so that the MTRS can automatically assign the correct verdict. Figure 6 shows a possible implementation of the test case.

The first eight lines are comments and meta-data for the test case. The next ten lines describe the actions to set up the test gateway correctly. Lines 19ff. describe the actual test case. Step 1

```
1  /**
2   * @Id 570
3   * @Version 1.1
4   * @Purpose Replying to an DEM Connection Information with Scope "ANNOUNCE" via unicast.
5   * The test is passed if the MTRS receives a DCI with Scope "REPLY".
6   * @SelectionCriteria Low
7   * @Keywords DCI
8   */
9  testCase MSLT1MTRS_M101 {
10     // the setup part
11     Component udpMan = create UDPMan(getSut().getIpAddress(), getSut().getUdpPort());
12     Component validator = create ConnectionInfoValidator();
13     link(udpMan, validator, "ConnectionInfoXML");
14     link(validator, udpMan, "ConnectionInfoXML");
15     OwnConnectionInfo ownDCI = getOwnConnectionInformation("449000001", "MTRS COY",
16                                          getSut().getTcpPortGateway1());
17     cm.start();
18
19     request("Please start your C2IS and be prepared to receive a DCI to " +
20             "which the System Under Test should reply.");
21
22     [!] send(ownDCI) to validator;
23
24     [?] receive(ConnectionInfo dci) from validator in 5000 {
25         assertEquals("NodeId", getSUT().getNodeId(), dci.getNodeId());
26         assertEquals("IpAddress", getSUT().getIpAddress(), dci.getIpAddress());
27         assertEquals("TcpPort", getSUT().getTcpPort(), dci.getTcpPort());
28         if (dci.getScope().equals(Scope.ANNOUNCE)) {
29             logInfo("Ignoring DCI with scope set to ANNOUNCE.");
30             repeat;
31         }
32     }
33     return Verdict.Pass;
34 }
```

Figure 6: MTRS Test Script

of the test case, the sending of a DCI, is done in line 22. Afterwards, in line 24, the MTRS expects to receive a DCI in 5000 ms. This means that the test case will fail if anything else is received in that time frame. If nothing was received within five seconds, the test will also fail. Lines 25-27 check if the fields of the DCI are set correctly. As DEM B is allowed to send its own DCI with scope ANNOUNCE at any time, lines 28-31 check this case and ignore such messages, i.e. the MTRS is waiting for the next DCI. If no error was found in the previous tests, line 33 sets the verdict of that test run to *pass*. For a more detailed description of the test language used by the MTRS, see Bau et al. (2007a). A user session with a successful test run is shown in figure 5.

5 Lessons Learned (So Far)

In a commercial setting, the acceptance of a testing facility strongly depends on its neutrality. From the very beginning, we have emphasized that the MTRS is not meant as a competitor to existing national C2ISs but has a clear focus on testing. Accordingly, the MIP-specific components were designed with error diagnosis rather than performance considerations in mind.

The timely availability of a test system is crucial. Ideally, it should be operable before all other systems. We have started work on the MTRS at an early stage. This made it possible to evaluate the draft specifications while they were being written. Based on the hands-on experience, various corrections and improvements could be made to the specifications. These changes have resulted in a disambiguated, simplified, and more robust MIP solution. The price to pay was that significant parts of the MTRS MIP gateway implementation had to be rewritten several times during the specification process.

5.1 Test Run Information

The MTRS gives full control to the test operator over his test results. The test operator may delete the results and logs for any test case or group at any time. Deletion may cover all test runs or only the outdated ones. (However, it is not possible to delete information on new, failed test runs, thus inhibiting that prior test results are restored.)

Interestingly, this feature is not used as often as one might expect. One possible explanation is that testers have a legitimate interest in keeping bad results as long as they do not occur too often. A test case, for which the test results have been removed, is indicated as untested (verdict *none*) both in the national and in MIP test report. This would imply that the test has not even been tried, whereas a failed test at least indicates some testing activity.

Information on failed tests also helps to find bugs in the MTRS test scripts and to identify and resolve ambiguities in the MIP specification. For that purpose, the MTRS client allows the test operator to provide comments directly in the MTRS client. These comments also appear in the official MIP test reports. Experience has shown that – depending on the complexity of the test case – it may take several iterations until a test script is robust enough to handle all border cases.

5.2 Regression Testing

An issue that gains more and more attention during the software development process is regression testing. If the implementation of a C2IS changes, the test operator may want to re-run all or a subset of the test cases. According to our server logs, for some MIP implementations, the total number of regression tests is much higher than the number of test runs needed to pass the test case for the first time (see figure 8). However, from a psychological point of view, it may be demotivating if someone has to delete all existing test results in order to get an overview of what tests need to be re-run. In particular, this holds true in the context of an international program, in which the test results are published within the community.

The MTRS supports regression testing by keeping track of the current C2IS version for each test run. The C2IS version is provided by the test operator and can be changed over time. If the test operator activates the regression testing feature, the MTRS client suppresses the results of all test cases that were not performed with the current version of the C2IS. Moreover, the outdated tests are highlighted in a dialog.

This feature is implemented in the MTRS Client only and can be deactivated at any time by the test operator. Neither does it affect existing test results, nor does it change the generated MIP test reports. I.e., the official MIP test report always includes the latest test verdicts, regardless of the C2IS version. Therefore, it is possible to perform regression testing without having to bother about the loss of all previous test results.

5.3 Test Versioning

An important aspect in the context of an international program like MIP concerns test case versioning. Ideally, test specifications should be written once and remain stable during the testing process. The same should hold for the executable test scripts derived from the MIP test specifications.

However, in practice, this requirement cannot be met. Test specifications are working documents that must be changed if the underlying specification changes or a bug is discovered in the test data. Likewise, test scripts are small pieces of software whose problems cannot be detected and eliminated completely before they are actually executed.

Keeping track of all modifications over time can be a great challenge, particularly, if the test operator is confronted with a huge pile of test documents but only sparse change logs.

In the MTRS, this problem has been addressed by labeling all MTRS test cases with a version number. Each test case has a major and a minor version. The minor version number is incremented whenever a change is made that does not require re-execution of previously successful tests but may allow C2ISs to pass a test they have failed before. The major version number is incremented if the test case has been extended, has changed its semantics, or the test evaluation has become stricter. In this case, all C2ISs have to re-run the test case.

The evaluation of the test case version number is tightly integrated in the MTRS client. When starting a new session, the test operator automatically receives information on all test cases that require re-assessment. Moreover, the verdict of outdated test results is set to *outdated* and the corresponding test cases are marked as outdated in the test suite overview and in the test reports.[3] Finally, the MTRS draws the user's attention to comments that may no longer be valid.

This way, test case versioning ensures that all C2ISs make use of the latest test specifications. We consider this as one of the key features to improve the test management.

[3]Initially, all outdated test results were reset to verdict *none*. Consequently, after a minor test case change, all failed tests were treated as if they had not been executed at all. However, we had to realize that some testers strongly prefer verdict *fail* or *outdated* to verdict *none*, because it documents some testing activity.

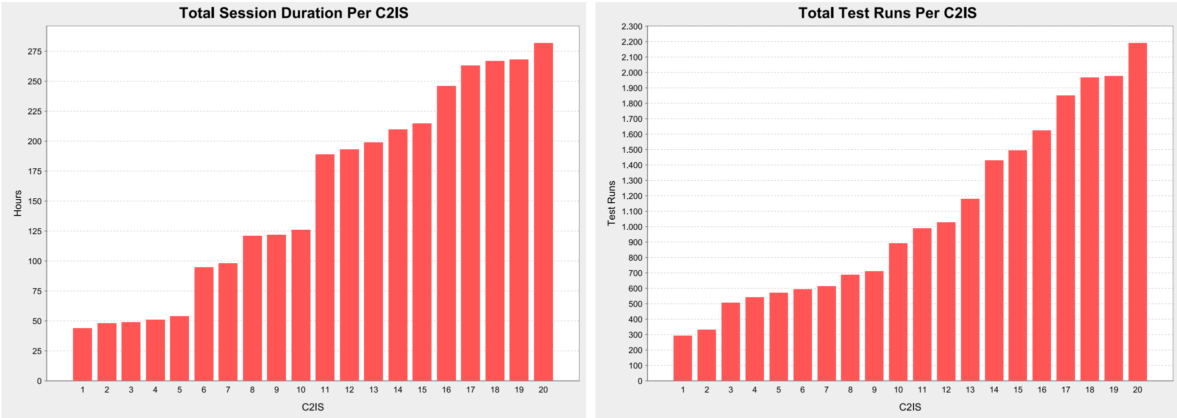

Figure 7: MTRS Utilization Grouped by C2IS

Test Verdict	**Number of Tests**	**Duration (hh:mm:ss)**	**Average Duration (mm:ss)**
Pass	9457	291:39:16	01:51
Inconclusive	3904	325:31:23	05:00
Fail	8122	340:04:57	02:31
$\sum$	21483	957:15:36	02:40

Table 1: Test Runs Grouped by Test Verdict

5.4 Utilization Statistics

As mentioned above, the SLT 1 tests have started in September 2007. Since then, 20 different MIP implementations have been tested with the MTRS.[4] As of March 14th, 2008, most systems have successfully completed SLT 1 (in total, only 105 out of 2140 SLT 1 tests are pending). In addition, about two thirds of all SLT 2 tests have been passed in average.

Figure 7 shows the amount of time spent on testing with the MTRS and the number of test cases which have been executed so far. The data is based on anonymized server logs and grouped by C2IS. In total, the test operators were logged into the MTRS for more than 3152 hours (approx. 394 man days), where idle times at the end of a session (because the test operator forgot to log out at the end of the day) have already been sorted out. During these sessions, more than 21,400 test cases have been executed. As shown in figure 7, there are several systems for which more than 1,800 test runs have been started. Given the fact that there are only 107 SLT 1 and 166 SLT 2 test cases, this means that, in average, every test case was executed six to seven times for these systems!

In table 1, the same results are grouped by test verdicts. Presently, the number of tests with verdict *fail* and *inconclusive* outweighs the number of successful tests. However, as the MIP

[4] Please note that several nations share the same MIP gateway implementation or even the complete C2IS. Accordingly, the number of C2ISs participating in SLT 3 and the number of nations supporting MIP Baseline 3 will be significantly higher.

protocol specification, the test specifications, the MTRS test scripts, and the system implementing and testing against the specification get more and more mature, regression testing will continuously improve the ratio of passed and failed tests.

Please note that table 1 does not imply that half of all functionality is presently broken; in fact, the systems have already passed most tests successfully. Instead, the table indicates that the MTRS allowed to identify problems that might have remained unnoticed in the (tested) C2IS otherwise. Moreover, it shows that successful tests are executed fastest. We assume that the longer test runs were caused by manual intervention, which tends to be a source of trouble. Finally, we can see that test execution takes about 30% of the overall testing time (total session duration / total test duration).

The fact that the MIP (test) specifications and the MTRS test scripts have changed over time, raises an interesting question: Which impact did these factors have on the testing efforts, or, more drastically, what overhead was caused by the MTRS? Although this question cannot be answered precisely, the server logs allow us to make a rough estimation. For each test run, the server keeps track of the C2IS (in anonymized form), the test result, and the version of the test case script. In a conservative approach, we can say that a test run for a given C2IS was in vain if

- its verdict is *fail* or *inconclusive* and
  - there is no test run with verdict *pass* and the same test case version, and
  - there is a test run with verdict *pass* and a higher test case version, which is the *first* test run for this test case version, and no other test runs have been executed between the two test runs.
- its verdict is *pass* and
  - there is another test run with a higher *major* test case version.

Following a more pessimistic assumption (from the point of view of MIP/MTRS), a test run was in vain if

- its verdict is *fail* or *inconclusive* and
  - the C2IS passes the same test with any higher test case version (minor or major) but not with the same one.
- its verdict is *pass* and
  - there is another test run with a higher *major* test case version.

Depending on whether the optimistic or pessimistic assumption is applied, 93.6% or 87.0% of all test runs were successful (and run against the latest test case version) or their failure can be traced back to a failure of the system under test.

Table 2 provides revised statistics based on the pessimistic assumption, where all useless test runs have been eliminated and the results are split up into SLT 1 and SLT 2. Despite the

Test Verdict	SLT 1		SLT 2	
	# Tests	Duration (Avg)	# Tests	Duration (Avg)
Pass	5239	01:14	3359	02:58
Inconclusive	1859	03:15	1208	08:29
Fail	4407	02:11	1928	03:24
$\sum$	11505	01:55	6495	04:08

Table 2: Test Runs Grouped by Test Verdict (based on revised data)

data cleanup, there is still a large number of test runs with verdict *inconclusive* or *fail* . This indicates that a lot of bugs have been identified and fixed in the C2IS during the testing period. Not surprisingly, the average duration of SLT 2 tests is significantly higher than the average duration of the SLT 1 tests. This is because large transactions with up to thousands of data records are replicated in SLT 2.

As stated above, regression testing is an important means to ensure correctness while the C2IS are being developed. With regard to the MTRS logs, we can define regression tests as follows:

A test t_r is a regression test if was run *after* another test t_2 where the verdict of t_2 is *pass* and the major versions of $t_r$and t_2 are identical.

An snapshot of the on-going regression testing efforts is depicted in figure 8. It demonstrates that the MIP community takes its testing efforts seriously.

6 Summary and Outlook

For Baseline 3, the MIP community spends tremendous resources on testing in order to achieve their vision of multilateral interoperability. In comparison to the former baseline 2, the number of tests has been increased by a factor of 5 for SLT 1 and a factor of 8 for SLT 2. At the same time, the test specification process and the quality of the individual tests have been improved significantly.

The FKIE has developed the MTRS as a tool for MIP conformance testing. Its purpose is to improve testing efficiency and to ultimately increase the quality and interoperability of MIP-compliant C2IS. The MTRS performs functional black box tests and makes no assumptions about the (user) interfaces of the tested C2IS apart from those required by the MIP specifications.

The MTRS is able to perform tests on different layers (network protocols, JC3IEDM database, C2IS integration) and with varying test configurations. At the time of writing this paper, all official MIP SLT 1 and SLT 2 test cases have been transformed into executable test scripts. Presently, 107 SLT 1 and 166 SLT 2 test scripts are available, many of which can only be performed with the MTRS, as they cover scenarios not reproducible by a regular C2IS. For SLT 3, the MIP community is in the process of specifying tests. Corresponding test scripts will be available on the MTRS server in summer 2008.

Figure 8: Regression Testing (based on revised data)

The client-server architecture of the MTRS makes it possible to maintain all test scripts in a central repository. This ensures that tests are always based on the latest scripts. Multiple C2ISs and test operators are able to run tests in parallel. Whenever a new test case (or group) is started, one or two MIP gateways are set up on the MTRS server at run-time. These gateways are assigned exclusively to a specific test operator and technical provisions ensure that there is no interference among them. Detailed protocol logs, which describe the information flow inside the test server gateway(s), allow for simplified test evaluation and error diagnosis. Moreover, the MTRS is able to export MIP test reports for national use and for the MIP test controllers.

The MTRS is offered free of charge to all national systems participating in the MIP System and Operational Level Tests. We try to make the MTRS as ”transparent” as possible by publishing the MTRS architecture and test cases, and by revealing the internal data flow at run-time. Server statistics indicate that the MTRS is used extensively by the MIP community. By Mid

March 2008, more than 21,400 SLT 1 and 2 tests have been executed with the MTRS.

Conformance testing with the MTRS and bi-/multilateral interoperability testing are complementing one another. Although the MTRS and its test scripts make it possible to discover all kinds of implementation errors, the set of test cases, for obvious reasons, can never be exhaustive. It fact, during bilateral testing, unforeseen problems were discovered that have finally resulted in new MTRS test scripts.

Further information on the test system can be found on the MTRS web server (`https://trac.fkie.fgan.de/MTRS`) that hosts a Wiki system with detailed documentation on the MTRS (including Flash tutorials) and a ticketing system that can be used for reporting defects, asking questions, etc. All test-related MIP documents as well as all MTRS test scripts are put under version control in a repository and can be downloaded from the website.

In future, we plan to enrich the MTRS by coverage analysis. For instance, it would be interesting to know which parts of the JC3IEDM (entities, attributes, domain values) have actually been covered during the MIP tests. We are also exploring ways to validate the data exchange between two C2IS during the MIP OLT. For instance, the MTRS could act as a man-in-the-middle. Alternatively, the MTRS may analyze the network traffic (retrieved by network sniffing) in an off-line/passive mode. Finally, our findings and tools could be generalized such that they become applicable to other interoperability programs and standardization efforts.

References

Bau, N., Gerz, M., and Glauer, M. (2007a). Ensuring Interoperability of C2 Information Systems – New Ways to Test Conformance to the MIP Solution. In *Military Communications and Information Systems Conference (MCC 2007)*, Bonn, Germany. Ministry of Defence, Germany, and NATO Assistant Secretary General for Defence Investment. 13

Bau, N., Gerz, M., Glauer, M., and Schüller, H. (2007b). Multilateral Interoperability Programme – Advancements in MIP Baseline 3. In *Twelfth International Command and Control Research and Technology Symposium (12th ICCRTS)*, Newport, RI, USA. Office of the Assistant Secretary of Defense, Command and Control Research Program (CCRP). 3

IEEE – Institute of Electrical and Electronics Engineers (1998). IEEE 1320.2 Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject). `http://electronics.ihs.com/document/abstract/FCSTFBAAAAAAAAAA`. 3

MIP – Multilateral Interoperability Programme (2006). MIP Standard Briefing. `http://www.mip-site.org`. 2

MIP – Multilateral Interoperability Programme (2007a). MIP Home Page. `http://www.mip-site.org`. 2

MIP – Multilateral Interoperability Programme (2007b). MIP Technical Interface Design Plan (MTIDP), Edition 3.5. `http://www.mip-site.org`. 11

MIP – Multilateral Interoperability Programme (2007c). The Joint C3 Information Exchange Data Model (JC3IEDM Main). Edition 3.1b. `http://www.mip-site.org`. 3

MIP – Multilateral Interoperability Programme (2007d). The MIP Test and Evaluation Master Plan (MTEMP). Edition 3.1. `http://www.mip-site.org`. 5

Nomenclature

C2IEDM	Command and Control Information Exchange Data Model
C2IS	Command and Control Information System
DCI	DEM Connection Information
DEM	Data Exchange Mechanism
FKIE	Research Institute for Communication, Information Processing, and Ergonomics
ILT	Implementation Level Tests
JC3IEDM	Joint Command, Control, and Consultation Information Exchange Data Model
MIP	Multilateral Interoperability Programme
MTIDP	MIP Technical Interface Design Plan
MTRS	MIP Test Reference System
OIG	Operational Information Group
OLT	Operational Level Tests
PDU	Protocol Data Unit
SLT	System Level Tests