Meeting notes for March 23, 2004 I++ DME implementer's conference call

Meeting secretary: John Horst
The National Institute of Standards and Technology (NIST)
Telephone: 301.975.3430
Email: john.horst@nist.gov

March 23, 2004

Name	Organization	Present
Ray Admire	Lockheed Martin	
Manfred Becker	Zeiss	√
Perluigi Borgogno	Wilcox Associates	
Joe Falco	NIST	√
Swen Haubold	Mitutoyo	
John Horst	NIST	√
René Keller	Metromec	√
Tom Kramer	NIST	√
Chuck Leckenby	Tecnomatix	
Mike Martini	General Electric Transportation	
Günter Moritz	Messtechnik Wetzlar	
Michel Penlae	Wilcox Associates	
Chiratana Pot	Tecnomatix	√
Walter Punegam	Zeiss	
Josef Resch	Zeiss	√
Bill Rippey	NIST	√
John Rosser	Renishaw	
Ken Sheehan	Entelegence	√
Dave Smith	LK	√
Rob Stewart	Daimler-Chrysler	√
Keith Stouffer	NIST	
Tim Taylor	General Electric Transportation	
Mark Vinson	Boeing	
Bob Waite	Daimler-Chrysler	
Betsy Weddendorf	General Electric Transportation	
Martin Wimmer	Zeiss	√

1 Discussion on action items from previous meeting

1. *To address the problem of differing machine coordinate limits and ranges for differing real CMM systems, NIST will build a utility that translates and/or rotates coordinate values in command files and what ever else is necessary. NIST also plans to allow the test suite user to enter a tool name through this additional utility.*

   No work on this yet.

2. *NIST plans to add the ability to automate test file execution to the client-side utility.*

   No work on this yet.

3. *NIST also plans to add several useful preset commands in separate windows (or as a pulldown menu), such as StopSession, EndSession, ClearAll, and Rewind, to the client-side utility.*

   No work on this yet.

4. *NIST will add more detail about socket errors in its reporting of errors on the client-side test utility.*

   No work on this yet.

5. *NIST plans to deliver a listing of server-side test cases prior to the next conference call for discussion then.*

   Delivered to implementors.

2 Interoperability demonstration at IMTS

John mentioned that NIST has secured (for the AIAG MEPT) a 9 m x 3 m booth at the IMTS in September, 2004 for the demonstration of metrology system interface standards (including I++ DME) using real CMMs. NIST needs to plan an effective demonstration of successful interoperability at the DME interface using I++ DME. A demo scenario needs to be defined. The initial idea is to have a few (two or more) CMMs and a few client-side software applications demonstrate interoperability across the DME interface in real-time on the shop floor. There would be a single PC laptop at each CMM. Each laptop would be equipped with the ability to connect to that particular CMM and run the same DMIS program on each client software in sequence. It also would be good to show the operation of the test utilities, in order to demonstrate that we are serious about not creating point-to-point solutions, but strive for actual interoperability. There are several options here. All involve the modification of the client-side implementation.

1. Modify client software to generate a standard log file (we suspect that many of you are doing this already) and NIST will generate a log file parser that will read this log file and from it generate an I++ DME command file and the NIST server-side utility will execute this file (responses go nowhere)

2. Modify client software to output a file of I++ commands at the end of the execution of each DMIS program and the NIST server-side utility will execute this file (responses go nowhere)

3. Modify client software to communicate command information to multiple sockets one is the real CMM and the other is the NIST server-side utility (and the response comes only from the real CMM)

These options were discussed and option one seemed agreeable to all. John asked that client-side implementors send their log files to NIST so we can advise the group as to the best way to proceed. Tom pointed out that we have two ways to proceed under option one, namely, develop a different log file parser for each log file format or specify a log file format that is agreeable to all and require everyone to switch to that common format (the latter requiring only one type of log file parser).

3 Status and update on I++ DME specification

John asked Josef what the schedule is to finish defining all remaining functionality, namely, camshaft/crankshaft? Josef said they have changed the previous schedule to add optical sensor first.

Josef said that the work on the specification is moving somewhat slowly, waiting for the upcoming meeting. Version 1.41 is being worked on.

Josef discussed the fact that the I++ DME spec writers have decided to allow wider autonomy within the server-side of the interface, meaning that certain actions can be performed without being commanded by the client, actions such as requesting a probe change from the server side or setting certain parameters.

4 Status and update on I++ DME implementations

There was insufficient time to discuss this.

5 Status and update on I++ DME test suite

John reviewed the concept and challenges of server-side test cases. Client-side test cases have been a part of our test suite from version 1.0 to the current version 2.1. However, we have really only had one server-side test case up until now, and that test case is "send correct and reasonable responses to all I++ DME commands." As NIST envisions it, server-side test cases should be distinctly different from client-side test cases. Client-side test cases are simply files of I++ DME command strings accompanied by test artifacts as necessary. On the other hand, to have server-side test cases consist of a set of response files would be appropriate only if we were to require that all client implementations send the same list of I++ DME commands for each test case, which would require an additional "test case" on the client-side consisting of a high level program and a requirement to interpret that program into I++ DME in only one way. Such a requirement is not part of the spec and need not be a part of the spec.

Such an approach is obviously too constraining. John argued that a better option is to allow the client-side implementor to define and execute any program they wish and select from a large set of appropriate test cases built into the server-side utility code. These responses will be fully deterministic (predictable behavior) and never random or uncertain. This approach will allow flexibility in the way a client can translate high level programs into I++ DME and yet still test to see if the client-side implementation responds appropriately to all or virtually all the different types of responses, both legal and illegal, that might occur.

Joe described the preliminary set of server-side test cases that was sent to the group last Friday. These have yet to be implemented and NIST merely wished to get some feedback from the group on the concept and details.

Dave suggested that NIST add position error (machine limit) as a radio button since it is not dependent on a particular command. Joe agreed that there should be a distinction made between those test cases that are independent of a particular command and those that are not.

Martin and René pointed out that the I++ DME spec does not say what to do when a server error occurs. This issue was discussed for some time and the conclusion seemed to be that that it is acceptable for this to be left up to the discretion of each client implementor and we should not request that the spec writers add such detail to the spec.

Rob said that a machine may give position error that is effectively the lag error. This distinction needs to be explicit. He also suggested that it would be good to have a help button connected to the portion in the spec. Dave said help button could have references and definitions.

Ken stressed that we need to have a precise definition of the meaning of an error. Much discussion ensued about the need for errors to be tightly defined in the spec. Tom and John argued that the current specification has ambiguous or non-existent definitions of the meanings of errors. They reiterated the need for the spec writers to fix this problem in the specification.

6 New and outstanding action items

- Client-side implementors will send example log files to NIST to aid NIST in development of a log file -> I++ DME command file parser.
- Advance the server-side test utility to include various test cases
- To address the problem of differing machine coordinate limits and ranges for differing real CMM systems, NIST will build a utility that translates and/or rotates coordinate values in command files and what ever else is necessary. NIST also plans to allow the test suite user to enter a tool name through this additional utility.
- NIST plans to add the ability to automate test file execution to the client-side utility.
- NIST also plans to add several useful preset commands in separate windows (or as a pulldown menu), such as StopSession, EndSession, ClearAll, and Rewind, to the client-side utility.
- NIST will add more detail about socket errors in its reporting of errors on the client-side test utility.

Our next meeting is planned for April 5, 2004. Talk with you then!