

**Brief Comparison of:
I++ DME Interface with
DmisEquip Module of DMIS Part 2**

**July 2-3, 2002
Frankfurt, Germany**

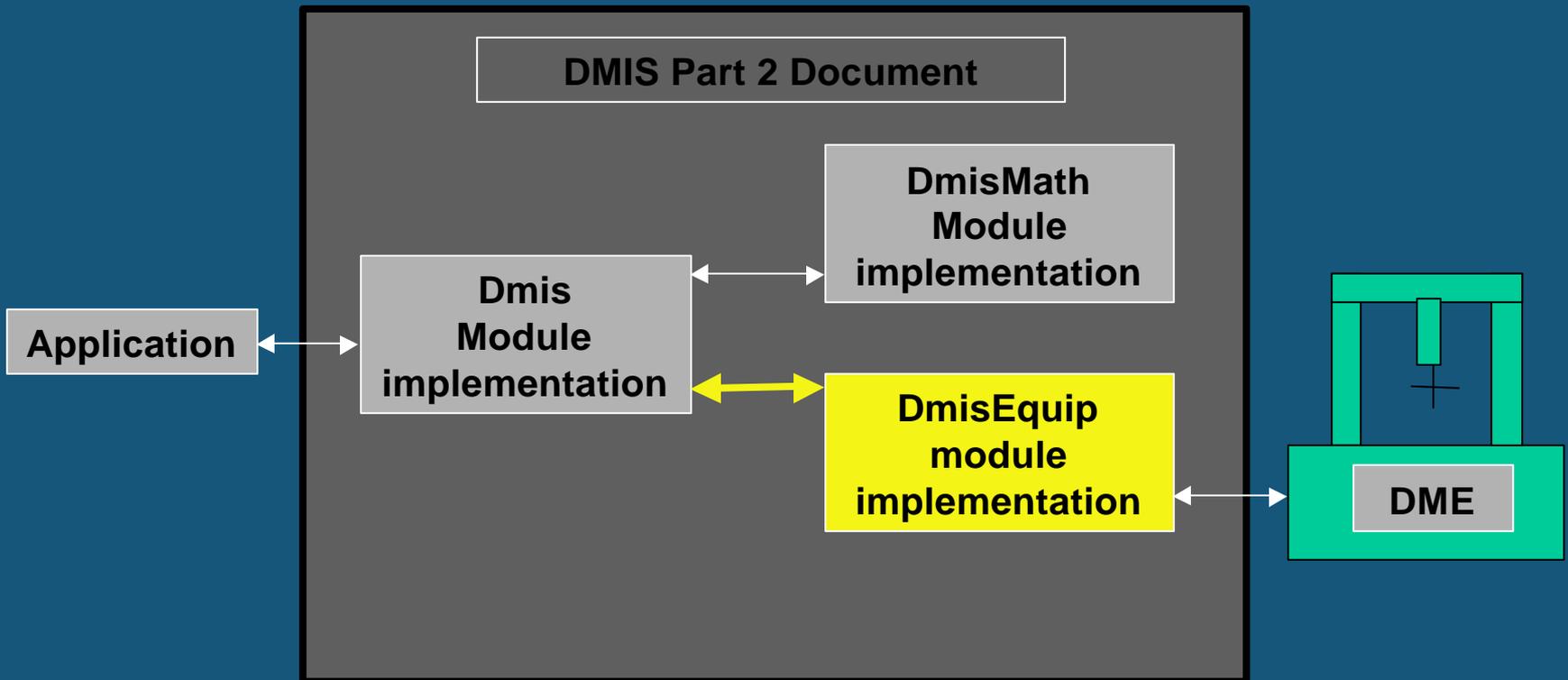
Thomas Kramer
National Institute of Standards and Technology (USA)

Outline

1. Overview of DMIS Part 2
2. Similarities (I++ and DmisEquip)
3. Differences (I++ and DmisEquip)
4. Questions and Comments

1. Overview of DMIS Part 2

DMIS Part 2 Architecture



About DMIS

1. DMIS = Dimensional Measuring Interface Standard
2. Developed by CAM-I
3. Only international standard language for programs for dimensional measuring equipment
4. At version 4.0
5. Now called DMIS Part 1

About DMIS Part 2

1. Dmis module implements DMIS Part 1
2. DmisEquip and DmisMath modules new
3. DmisEquip module comparable to I++ DME
4. Being developed by CAM-I
5. Specifies functionality and interface
6. Object-oriented (written in CORBA IDL)

2. Similarities

of I++ DME Interface and DmisEquip module

Command Set Similarities

both have commands for

measuring points

moving in free space

selecting and changing sensors

getting and setting probing parameters

getting and setting free space motion parameters

getting and setting sensor parameters

enabling manual probing

Other Similarities

both are

used for low-level DME control
object oriented
works in progress

3. Differences

**between I++ DME Interface
and DmisEquip module**

Messaging Differences

I++ DME Interface

uses sockets to send strings
needs message syntax
uses event messages

DmisEquip

uses ORB for remote calls
needs no message syntax
uses no events

Command Set Differences

I++ DME Interface

- does not support all of DMIS Part 1
- no scanning commands
- one server for each column
- no non-touch sensor commands
- no rotary table commands
- no calibration commands
- has coordinate system commands

DmisEquip

- supports all of DMIS Part 1
- has scanning commands
- one server handles multiple columns
- has non-touch sensor commands
- has rotary table commands
- has calibration commands
- no coordinate system commands

Other Differences

I++ DME Interface

real-time implementation possible

application is client

modeled in UML, C++, ad hoc

is self-contained

DmisEquip

real-time implementation not possible

Dmis module is client

modeled in CORBA IDL

not self-contained (needs many items from Dmis module)

4. Questions and Comments

