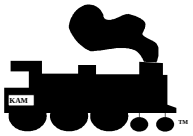
Jacobsen Declaration Exhibit AL

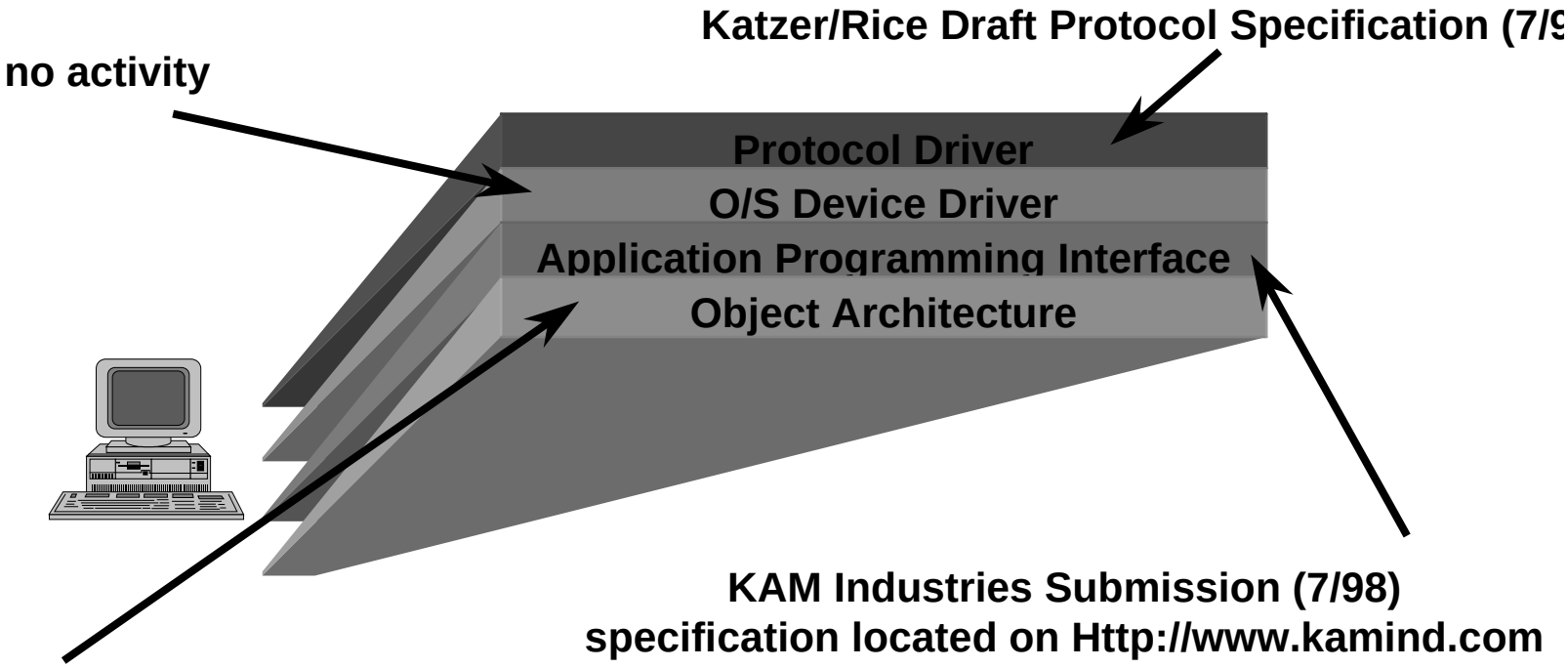
NMRA Software Architecture Status

Matt Katzer
KAM Industries
Portland, Or.



Status of NMRA Application S/W Architecture Model

- **There are four parts to the NMRA DCC software architecture model**



Status of NMRA Application S/W Architecture Model (cont.)

- **Protocol Level**
  - **hardware Products**
    - » North Coast Engineering, Wangrow Electronics
    - » Easy DCC
    - » ZTC systems
  - **Software drivers for command station hardware**
    - » WinLok, Engine Commander®, Railroad Company Tayden Design
  - **Generic draft protocol driver**
    - » Engine Commander®

Status of NMRA Application S/W Architecture Model (cont.)

- **Device Driver Level**
  - » no activity
- **Application Interface Level**
  - – **hardware Products**
    - » not applicable to hardware
  - – **Microsoft COM/DCOM implementation of API**
    - » Engine Commander®
    - » Computer Dispatcher® (March 98)
    - » Generic type library available for linking with application written in Java, Visual Basic, C/C++
  - – **CORBA support**
    - » no activity

Status of NMRA Application S/W Architecture Model (cont.)

- **Object level**
  - **Rosa application model proposed (update on http://www.digi-toys.com)**
  - **hardware Products**
    - » not applicable to hardware
  - **Software products**
    - » Engine Commander® and Train Server® conforms in architecture model
  - **COM support**
    - » no activity
  - **CORBA support**
    - » no activity

API command summary

- **API Command classes**
  - CV
  - Engine
  - Consist
  - Accessory
  - Command
  - Programming
  - Communications
  - Command
  - Decoder
  - Cab
  - Feedback
  - Callback methods

These are the major classes of commands needed in most DCC software applications.

We have implemented Engine Commander® and are in the development phase of Computer Dispatcher®

•Train Tools API

- **Fucntions**
- **DccCVGetValue();**
  DccCVSetValue();
  DccCVGetStatus();
  DccCVSetStatus();
  DccCVGetName();
  DccCVGetMaxRegister();
  DccCVGetMinRegister();
- **Accessory Commands**

  DccAccGetFunction( );
  DccAccSetFunction( );
  DccAccGetFunctionAll( );
  DccAccSetFunctionAll( );
  DccAccGetFunctionMax( );
  DccAccGetName( );
  DccAccSetName( );
  DccAccGetFunctionName( );
  DccAccSetFunctionName( );

Train Tools API (cont.)

- **Engine**

  DccEngGetSpeed( );
  DccEngSetSpeed( );
  DccEngGetFunction( );
  DccEngSetFunction( );
  DccEngGetFunctionMax( );
  DccEngGetName( );
  DccEngSetName( );
  DccEngGetFunctionName( );
  DccEngSetFunctionName( );
  DccEngGetSpeedSteps( );
  DccEngSetSpeedSteps( );

- **Consist**

  DccEngConsistGetMax( );
  DccEngConsistSetParent( );
  DccEngConsistAddUnit( );
  DccEngConsistRemoveUnit( );
  DccEngConsistGetParent( );

Train Tools API(cont.)

- **Command Station**

  DccOprGetStationStatus();
  DccOprTurnOnStation();
  DccOprStartStation();
  DccOprClearStation();
  DccOprStopStation();
  DccOprPowerOn();
  DccOprPowerOff();
  DccOprHardReset();
  DccOprEmergencyStop();

- **Programming**

  DccProgramGetStatus();
  DccProgramSetMode( );
  DccProgramGetMode( );
  DccProgramWriteCV( );
  DccProgramReadCV( );
  DccProgramWriteDecoderToDataBase( );
  DccProgramReadDecoderFromDataBase( );

Train Tools API(cont.)

- **Communications**

  DccProgramGetStatus();
  DccProgramSetMode( );
  DccProgramGetMode( );
  DccProgramWriteCV( );
  DccProgramReadCV( );
  DccProgramWriteDecoderToDataBase( );
  DccProgramReadDecoderFromDataBase( );

- **Command**

  DccCmdCommand( );
  DccCmdConnect( );
  DccCmdDisConnect( );

- **Cab**

  DccCabWriteMessage( );
  DccCabReadMessage( );
  DccCabSetDccObject( );
  DccCabGetDccObject( );
  DccCabAdd( );
  DccCabDelete( );
  DccCabTranslate( );
  DccCabLookupDccObject( );

Train Tools API(cont.)

- **Decoder**

  DccDecoderGetMaxModels();
  DccDecoderGetModelName();
  DccDecoderGetMaxAddress();
  DccDecoderCheckAddrInUse();
  DccDecoderGetMfgName( );
  DccDecoderGetPowerMode( );
  DccDecoderAddAddr()
  DccDecoderGetModelFacility()
  DccDecoderReconnectObject( );
  DccDecoderChangeAddress( )
  DccDecoderTranslate( )
  DccDecoderSetModelToObject()
  DccDecoderGetMaxSpeed( );
  DccDecoderGetObjectCount()
  DccDecoderGetObjectAtIndex()
  DccDecoderDel( );
  DccDecoderGetErrorState( )

Train Tools API(cont.)

- **Feedback**

  DccFeedbackErrorMessage( );
  DccFeedbackAccessoryBit( );
  DccFeedbackAccessoryAll( );
  DccFeedbackEngineResponse( );
  DccFeedbackCV( );
  DccFeedbackMessagesCab( );
  DccFeedbackMisc( );

- **Callbacks**

  DccResponseErrorMessage();
  DccResponseAccessoryBit();
  DccResponseAccessoryAll();
  DccResponseEngineResponse();
  DccResponseCV();
  DccResponseCabMessage();
  DccResponseMisc();

Train Tools Api(cont.)

- **Time**

  DccMiscGetClockTime( );
  DccMiscSetClockTime( );

- **Command Station**

  DccMiscGetControllerName( );
  DccMiscGetControllerNameAtPort( );
  DccMiscGetCommandStationIndex( );
  DccMiscMaxControllerID( );
  DccMiscSetCommandStationValue( );
  DccMiscGetCommandStationValue( );
  DccMiscGetControllerFacility( );

- **Misc**

  DccMiscGetErrorMsg ( );
  DccMiscGetApiName( );
  DccMiscGetInterfaceVersion( );
  DccMiscSaveData( );

Questions ?

Matt Katzer
email: mkatzer@kam.rain.com
web: http://kam.rain.com
home: 503-291-1221

Legal Disclaimer

- **KAM Industries has submitted the Trains Tools® Application Programming Interface to the NMRA DCC Working group for RP approval under the following conditions;**
  - **If the API is ratified as a Reference Practice(RP) KAM will transfer copyright of the document to the NMRA, otherwise the document and API's remain KAM's copyrighted property.**
  - **If the API is transferred to the NMRA, KAM retains rights to publish and use the RP Train Tools API document in their product, website and documentation as appropriate without any license fees or restrictions.**