A Prototype
of the
CDF Online Hardware Database
(HwDB)

Geri Goeransson
CDF Online Computing
March 15, 1999

The CDF Online Hardware Database (HwDB) combines much of the functionality of the Rabbit (RDB), FASTBUS (FBDB) and ResMgr Database (ResDB) from the first run. Information about individual modules, crates, channels and components is contained within it.

A prototype of the HwDB has been created using the problem of building the readout lists for the Central Calorimetry as the premise for the design. The information mirrors that of the Calibration Database, but with different access keys.

The prototype for the HwDB was designed using Oracle Designer 2000 and accesses an Oracle 8 Database. The access routines are being written in Java using the JDBC for database connections. The work was done on B0DAU30, an SGI Origin running IRIX64-6.5.

<u>Central Calorimetry as a Test Data Set</u>

The 16 Central Crates, 2 VRB DAQ Crates, 16 Tracers, 96 ADCs and 1500+ r channels, and the two Central components, CEM and CHA comprise the data set. The information is housed in 4 relational tables: CARDS, CHANNELS, CRATES, and COMPONENTS. A discussion follows on the fields and queries to pull the information from the Oracle Database.

<u>HwDB Table Descriptions</u>

COMPONENTS is a 2-column table. The first column contains the key, component number (COMPNUMID). It is defined as an unsigned integer. The component name (COMPNAME) resides in the second column. It is defined as a 4-character string. Both of these columns must be defined for the row. The description of the COMPONENTS table and its attributes are

COMPONENTS TABLE :

Name	Null?	Type	Key?
COMPNUMID	NOT NULL	NUMBER	YES
COMPNAME	NOT NULL	VARCHAR2	

A typical SQL statement to insert a COMPONENTS tuple into the HwDB would be

```
INSERT INTO COMPONENTS (COMPNUMID, COMPNAME)
        VALUES (`CEM', 3);
```

CRATES is a multi-column table. The first column is the crate name (CRATENAME). It is stored as a character string. The crate number (CRATENUM) resides in column 2 and is defined as an unsigned integer. The crate type (CRATETYPE) is stored in column 3 as a character string. For the Central Crates, the type is set to "CENTRAL". For the two VRB Crates the type is set to "DAQVRB". The CRATENAME, CRATENUM AND CRATETYPE are all required fields and must be defined to store a row in the database. There are 2 optional fields in the 4th and 5th columns. The first is the Tracer slot (TRACERSLOT). It is stored as an unsigned integer. A crate may or may not have a Tracer in it. The other optional attribute is the JDKCLASS. This is the value of the JDK class for the Crate Class. It is stored as an unsigned integer value. This is the bridge between the Oracle Database and the Java class for the Crate. The primary key for CRATES is the CRATENAME. The description of the CRATES table is

CRATES TABLE :

Name	Null?	Type	Key?
CRATENAME	NOT NULL	VARCHAR2	YES
CRATENUM	NOT NULL	NUMBER	
CRATETYPE	NOT NULL	VARCHAR2	
JDKCLASS		NUMBER	
TRACERSLOT		NUMBER	

A typical SQL statement to insert a CRATES tuple into the HwDB would be

```
INSERT INTO CRATES
        (CRATENAME, CRATENUM, CRATETYPE, TRACERSLOT)
        VALUES (`CENTRAL_CRATE_0', 0, `CENTRAL', 2);
```

It is not necessary to include a value for the JDKCLASS as this is not a required attribute.

CARDS is also a multi-column table containing information on the cards stored in the crates. The first column has the name of the card (CARDNAME) stored as a 38 character string. CARDNAME is the primary key for the CARDS table. Column 2 is the CARD_ONLINE attribute. If a card is online, a 1 is stored in the CARD_ONLINE field. If not, a 0 is stored. The slot (SLOT) is found in column 3. This is the location of the card in the crate. It is defined as an unsigned integer. The 4th column in the table is card type (CARD_TYPE) and is defined as a 10-character string. Card types are defined for every card. The CARD_TYPE for a Tracer is "TRACER", for a Daq Vrb the CARD_TYPE is "VRB". The CARD_TYPE for Ad Mem modules is "ADMEM". The 5th column is the CRATENAME. This is the name of the Crate where this particular card resides. It is defined as a 38-character string. The CARDNAME, CARD_ONLINE, SLOT, CARD_TYPE and CRATENAME fields are required to store a row in the CARDS table and cannot be left undefined.

A number of constraints or requirements have been defined for certain module types. For the ADMEM cards, the channel count (CHANNELCT) and the component number (COMPNUMID) read out by the ADMEM card must be defined when the ADMEM record is stored. The CHANNELCT is an unsigned integer field. The COMPNUMID must be one of the values found in the COMPONENT table.

For the Tracer cards, the name of the VRB card (CARDNAME_T0) and the channel number (VRB_CONNECTION) attached to the Tracer must also be defined when the information on the Tracer card is stored in the table. The CARDNAME_TO must be the name of a Daq Vrb stored in the CARDS table. The VRB_CONNECTION is defined as an unsigned integer.

Optional fields for the CARDS table are PROM and JDKClass. The PROM field is the name of the prom found in the module. It is defined as a 38-character string. The JDKClass is defined as an unsigned integer value. The description of the CARDS table is

CARDS TABLE :

Name	Null?	Type	Key?
CARDNAME	NOT NULL	VARCHAR2	YES
CARD_ONLINE	NOT NULL	NUMBER	
SLOT	NOT NULL	NUMBER	
CARD_TYPE	NOT NULL	VARCHAR2	
CRATENAME	NOT NULL	VARCHAR2	YES
COMPNUMID		NUMBER	YES (ADMEM)
CHANNELCT		NUMBER	
PROM		VARCHAR2	
JDKCLASS		NUMBER	
VRB_CONNECTION		NUMBER	
CARDNAME_TO		VARCHAR2	YES (TRACER)

It is important to note that the constraints on the CARD_TYPE are not listed as NOT NULL in the table description. This is because the constraint is on a particular CARD_TYPE and not the entire table.

A typical SQL statement to insert a CARDS tuple into the HwDB would be

```
INSERT INTO CARDS
        (CARDNAME, CARD_ONLINE, SLOT, CARD_TYPE, CRATE_NAME,
        VRB_CONNECTION, CARDNAME_T0)
VALUES (‘TRACER_00’, 1, 2, ‘TRACER’, ‘CENTRAL_CRATE_0’, 1,
        ‘VRB_DAQ_00’);
```

The CHANNELS table contains information on the channels associated with cards in the CARDS table. Each row in the CHANNELS table is given a sequential channel number (CHANNEL ID). This uniquely identifies a row in the CHANNELS table. It is the primary key for the CHANNELS table. It is defined as an unsigned integer value. The CHANNELID is automatically defined by the Oracle Database upon storage of the row. The channel number as computed for a given channel is stored in the CHANNELNUMBER field. The CHANNELNUMBER is defined as unsigned integer value. In the 3rd column, is the Block number (BLOCK) within the raw data bank. It is defined as an unsigned integer value. The 4th column is the subaddress of the channel (SUBADDRESS). It is defined as an unsigned integer value. In the 5th column, is the CARDNAME. The CARDNAME contains the name of a card from the CARDS table containing the channel defined in this data record. The CARDNAME is a 38-character string. All of these fields are required and a value must be provided for each when the row is stored in the CHANNELS table. The last column is the JDKClass. This is optional. The description of the CHANNELS table is

CHANNELS TABLE :

Name	Null?	Type	Key
CHANNELID	NOT NULL	NUMBER	YES
CHANNELNUMBER	NOT NULL	NUMBER	
BLOCK	NOT NULL	NUMBER	
SUBADDRESS	NOT NULL	NUMBER	
CARDNAME	NOT NULL	VARCHAR2	YES
JDKCLASS		NUMBER	

A typical SQL statement to insert a CHANNELS tuple into the HwDB would be

```
INSERT INTO CHANNELS
        (CHANNELID, CHANNELNUMBER, BLOCK, SUBADDRESS, CARDNAME, JDKCLASS)
VALUES (CHANNELID_SEQ.NEXTVAL, 65, 0, 1, `ADMEM_01', 12345);
```

Queries

Queries into the HwDB can access any of the data contained in the CARDS, CRATES, CHANNELS or COMPONENTS tables. Queries can be made across multiple tables using the information contained in one table to search into another.

Typical queries to build the readout lists for the Central Calorimetry would be, generate a list of all the channels associated with a given component or generate a list of card slots for a given component. A number of example queries are given below.

A SQL query generating a list of all the channels associated with the CEM component from the prototype HwDB would be

```
SELECT CHANNELNUMBER, BLOCK, SUBADDRESS
FROM CHANNELS, CARDS, COMPONENTS
WHERE CHANNELS.CARDNAME = CARDS.CARDNAME
AND CARDS.COMPNUMID = COMPONENTS.COMPNUMID
AND COMPONENTS.COMPNAME = `CEM';
```

A SQL query generating a list of all the cards and their slots reading out the CHA components from the prototype HwDB would be

```
SELECT CARDNAME, SLOT  FROM CARDS, COMPONENTS WHERE
CARDS.COMPNUMID = COMPONENTS.COMPNUMID AND COMPONENTS.COMPNAME =
'CHA';
```

A SQL query used by programs looking at the event raw data would be to generate the crate, slot, subaddress knowing the component is the CEM, channel number is 1106 and block number is 5

```
SELECT CRATENAME, SLOT, SUBADDRESS  FROM CARDS, CHANNELS, COMPONENTS
WHERE CHANNELS.CHANNELNUMBER = 1106 AND
CHANNELS.CARDNAME = CARDS.CARDNAME AND
CARDS.COMPNUMID = COMPONENTS.COMPNUMID AND
COMPONENTS.COMPNAME = `CEM';
```

A SQL query

Open Issues

There are a number of issues as yet unresolved regarding the prototype HwDB. Timing studies need to be done with Oracle JDBC. The Oracle JDBC is thin-layer and what affect this will have on the access times is unknown.

There is also a question of whether all the data fields and attributes have been captured to generate readout lists for more than just the Central Calorimetry.

The HwDB is still evolving and changes probably will be made to accommodate larger data sets. However, with the prototype information the Oracle database product presents itself as a good candidate for the HwDB.