

Phenotyping Data Management System for IBDB V2

INTRODUCTION

The Data Management System (DMS) is the IBDB component of that manages germplasm characterization and evaluation data for genetic resources and crop improvement projects. DMS links these data to germplasm and pedigree information in the Genealogy Management System (GMS), to location information in the Location Management Module (LMM) and to information on genes, markers and alleles in the Genotyping Data Management System (GDMS) as well as providing links to other specialized data sources. DMS also allows integration of data from different characterization and evaluation studies, thus permitting a broad range of queries across trials or types of variables.

The Functions of the DMS are to:

1. Store and manage documented and structured data phenotyping data from germplasm characterization and evaluation studies as commonly conducted in crop improvement programs
2. Link data to specialised data sources such as GMS, soil and climate databases and
3. Facilitate queries, searches and data extraction across studies according to structured criteria for data selection as required by plant breeders.

All types of phenotyping data will be accommodated in DMS including raw observed data, derived data, means data and summary statistics. Data may have numeric values or character values or categorical class values. For example, observations on disease resistance or nutrient efficiency of a genotype can be numerical measurements, scored or calculated indices or character data.

As a general principle, any data that are routinely represented in field books or laboratory books or spread sheets will be accommodated in DMS. It will handle data and documentation on the basis of individual phenotyping studies which may contain data from many environments (trial instances), and from different sampling levels – study level, environment level, plot level, sample level. Integration across studies will be facilitated by the use of controlled vocabularies and ontologies managed by the Ontology Management System (OMS)

Structure of Phenotyping Data

In order to clarify the definition of entities in the DMS data model we will consider data from a fictional split-plot field experiment. Although this type of experiment is not commonly used in crop improvement it does contain all the elements of the variety of experimental designs which are used and so served as a good motivational model. Data sets are typically arranged in columns as in Table 1, we call the columns VARIABLES since each row may record a different value for each variable. The first seven variables define the source and context of the data. We refer to such variables as LABELS. The last three variables contain measured data. We call these data variables VARIATES, there are usually several variates in a data set. In general we can think of

the LABELS as variables for which we know the values before we do the experiment, and VARIATES, the ones we measure during the experiment.

Table 1 Serial Spread-sheet Representation for a Split-plot experiment from Study S9801

| | A | B | C | D | E | F | G | H | I | J | K |
|----|------|-----|----------|---------|-------|---------|-----|------|-------|-----|-----|
| 1 | PLOT | REP | MAINPLOT | SUBPLOT | ENTRY | VARIETY | GID | FERT | YIELD | PHT | BLB |
| 2 | 1 | 1 | 1 | 1 | 2 | B | 100 | 100 | 10.3 | 80 | 3 |
| 3 | 2 | 1 | 1 | 2 | 2 | B | 100 | 200 | 12.7 | 85 | 3 |
| 4 | 3 | 1 | 2 | 1 | 3 | C | 102 | 200 | 18.7 | 103 | 5 |
| 5 | 4 | 1 | 2 | 2 | 3 | C | 102 | 100 | 13.7 | 88 | 4 |
| 6 | 5 | 1 | 3 | 1 | 1 | A | 105 | 100 | 12.6 | 79 | 2 |
| 7 | 6 | 1 | 3 | 2 | 1 | A | 105 | 200 | 16.7 | 102 | 1 |
| 8 | 7 | 2 | 1 | 1 | 2 | B | 100 | 200 | 19.2 | 87 | 4 |
| 9 | 8 | 2 | 1 | 2 | 2 | B | 100 | 100 | 12.3 | 90 | 3 |
| 10 | 9 | 2 | 2 | 1 | 1 | A | 105 | 100 | 17.1 | 92 | 1 |
| 11 | 10 | 2 | 2 | 2 | 1 | A | 105 | 200 | 14.1 | 102 | 2 |
| 12 | 11 | 2 | 3 | 1 | 3 | C | 102 | 200 | 16.3 | 100 | 6 |
| 13 | 12 | 2 | 3 | 2 | 3 | C | 102 | 100 | 12.2 | 98 | 5 |

STUDIES

A STUDY is the basic, reportable unit of research, it is synonymous with the notions of experiment, nursery or trial. Since DMS must deal with any of these we will use the term study. A study may be characterized by a set of scientific objectives and testable hypotheses and results in the collection of one or more datasets similar to that in Table 1 or it may be simply a convenient package of research activities such as all the replicated field evaluation for a breeding program in a particular year. A study always has some metadata associated with it, such as its name, the PI, the institute, IP status and so on. These are variables, or more precisely labels which take a single value which applies to the whole study. We call them STUDY LABELS, and they often appear as headers to tables such as Table 1 above.

The division of data into sets is usually motivated by convenience, for example data collected from different sampling scales is most conveniently treated in different datasets. Similarly, data collected at different times or from different locations are also often treated as different data sets, although it is feasible and usually preferable to treat these divisions in a single dataset.

Each row in a dataset corresponds to an OBSERVATION UNIT of the study. Values of STUDY labels apply to all the observation units in a study (from any dataset in the study).

ANNOTATION OF VARIABLES

Variables are named and described freely by users, but consistently within each study. However they are annotated by terms from three controlled vocabularies:

- The PROPERTY which describes the context of the sampling unit and experimental material, if the variable is a label, or the trait being measured if it is a variate,

- the METHOD which describes how the PROPERTY is applied or the protocol by which a variate is measured and
- the SCALE which describes the units in which the label levels or variate values are recorded.

These three controlled vocabularies are terms in the Crop Ontology, and together they define the variables in the database. Every variable in the database is annotated by a combination of PROPERTY, METHOD and SCALE terms, and every unique combination of these terms occurring in the database defines a STANDARD VARIABLE. STANDARD VARIABLES are given standard names and descriptions in the ontology, but are referred to locally (within a study) by local names and descriptions assigned by the researcher. STANDARD VARIABLES link data across studies, and sets of STANDARD VARIABLES, for example those with the same property, or those with the same PROPERTY and METHOD or PROPERTY and SCALE link data about the same property across studies. All values of a particular STANDARD VARIABLE should be of the same data type. At present three types are being considered: numeric, character and database IDs (links to records in other database modules such as GIDs). Variables can also be categorical in which case they can only take on values from a defined set of VALID VALUES. This range can be extended to cover other types such as binary, picture, link or other object.

LABELS and LEVELS

LABELS are classifying variables in a study which take values from finite sets of discrete LEVELS. These levels document the source and context of the data by expressing the conditions under which the data were collected or derived. For example, the names of treatments or design structures applying to the unit or units from which the data are recorded, or conditions such as the time and location of measurement. These LABELS are usually listed in columns in the data set as in Table 1. The Study Name will be treated in the data model as a LABEL with exactly one level. Hence, every study has at least one LABEL.

In phenotyping experiments we can identify four groups of labels which describe different parts of the study – STUDY labels, LOCATION (or environment) labels, ENTRY (or germplasm) labels, and FIELD TRIAL (or design) labels. In the example in Table 2, rows 1 to 8 and 11 to 16 describe STUDY labels, rows 17-19 describe LOCATION labels, rows 26-28 describe ENTRY labels, and rows 22 to 25 and 29 describe FIELD TRIAL labels. Labels listed in the CONDITION section have only one level or value for the particular data table annotated by the description sheet. Labels listed in the LABEL section have multiple levels and correspond to columns in the observation table (Table 1). Combinations of one level from each label define the observation units – rows of a spreadsheet as shown in Table 1.

Table 2. Description Sheet showing the annotation of variables for the data shown in Table 1

| | A | B | C | D | E | F | G |
|----|------------|------------------------------|----------------|-----------------|---------------------|-----------|-----------|
| 1 | STUDY | S9801 | | | | | |
| 2 | TITLE | Study 1 of 1998 | | | | | |
| 3 | PMKEY | 123 | | | | | |
| 4 | OBJECTIVE | Test the schema | | | | | |
| 5 | START DATE | 20111202 | | | | | |
| 6 | END DATE | 20111203 | | | | | |
| 7 | STUDY TYPE | E | | | | | |
| 8 | DATA SET | PLOT DATA | | | | | |
| 9 | | | | | | | |
| 10 | CONDITION | DESCRIPTION | PROPERTY | SCALE | METHOD | DATA TYPE | VALUE |
| 11 | INSTITUTE | Study Institute | INSTITUTE | DBCV | CONDUCTED | C | GCP |
| 12 | INST ID | Study Institute ID | INSTITUTE | DBID | CONDUCTED | N | |
| 13 | PI | Principal Investigator | PERSON | DBCV | ASSIGNED | C | Artlet |
| 14 | PI ID | ID of Principal Investigator | PERSON | BCID | ASSIGNED | C | 1 |
| 15 | IPSTATUS | IP Status | IP | IP STATUS | ASSIGNED | C | 0 |
| 16 | RELEASE | Data Release Date | IP | DATE | ASSIGNED | N | 20121011 |
| 17 | TRIAL | Trial Number | TRIAL INSTANCE | NUMBER | ENUMERATED | N | 1 |
| 18 | SITE | Trial Site | LOCATION | DBCV | SELECTED | C | LOS BANOS |
| 19 | SITE ID | Trial Site ID | LOCATION | DBID | SELECTED | N | 10 |
| 20 | | | | | | | |
| 21 | LABEL | DESCRIPTION | PROPERTY | SCALE | METHOD | DATA TYPE | VALUE |
| 22 | PLOT | Plot number | PLOT NUMBER | NUMBER | ENUMERATED | N | |
| 23 | REP | Replication | REPLICATION | NUMBER | ASSIGNED | N | |
| 24 | MAINPLOT | Main plot number | BLOCK | NUMBER | ASSIGNED | N | |
| 25 | SUBPLOT | Sub-plot in main plot | BLOCK | NUMBER | ASSIGNED | N | |
| 26 | ENTRY | Entry Number | GERMPLASM ENTR | NUMBER | ENUMERATED | N | |
| 27 | VARIETY | Variety name | GERMPLASM IDEN | DBCV | ASSIGNED | C | |
| 28 | GID | Variety GID | GERMPLASM IDEN | DBID | ASSIGNED | N | |
| 29 | FERT | Fertilizer Level | N FERTILIZER | KG/HA | APPLIED | N | |
| 30 | | | | | | | |
| 31 | CONSTANT | DESCRIPTION | PROPERTY | SCALE | METHOD | DATA TYPE | VALUE |
| 32 | PH | Site Ph | SOIL PH | PH | PH METER | N | 6.3 |
| 33 | | | | | | | |
| 34 | VARIATE | DESCRIPTION | PROPERTY | SCALE | METHOD | DATA TYPE | VALUE |
| 35 | YIELD | Grain yield | GRAIN YIELD | KG/HA | HARVEST 5 SQM | N | |
| 36 | PHT | Plant height | PLANT HEIGHT | CM | SOIL TO PANICLE TIP | N | |
| 37 | BLB | BLB Resistance | BLB RESISTANCE | SES SCORE (1-9) | VISUAL ASSESSMENT | C | |

The role of a variable being a **CONDITION** or and **LABEL** is dependent on the scope of the data table. For example if data were collected from several sites, then the complete data set including data from all sites would have to have a label column indication the location from where the data for that row was collected so **SITE** would be a **LABEL**. IF you only show a part of the dataset coming from one location then **SITE** is a **CONDITION** for that data table.

VARIATES AND VALUES

VARIATES are the variables which contain the data observed in the experiment – the phenotypic data. They usually appear as columns in the data table as **YIELD**, **PHT** and **BLB** in Table 1. Variates which have only one value pertaining to all observation units in the data table are called **CONSTANTS**.

Note however, that the status of **CONDITION** and **CONSTANT** depends on the data shown. If data in Table 1 were for two locations then **SITE** would have to be represented as a **LABEL** in the data table (ie a column) and similarly for **PH**.

OBSERVATION UNITS

Data sources such as field objects or sampling units are identified by combinations of levels of design or sampling **LABELS**. In Table 1, the **LABELS** **REP**, **MAINPLOT** and **SUBPLOT** are all

design LABELS and combinations of one level from each identify physical sub-plots in the study. Other LABELS define the context of the data, in experiments these are called treatment LABELS. Combinations of one level from each treatment LABEL define the treatments which are applied to field objects. In Table 1 VARIETY and FERT are treatment LABELS.

Data values such as treatment means, as in Table 3, are associated with level combinations of treatment LABELS which do not correspond to field objects but which can be thought of as data sources. Both types of data sources, field objects and treatment combinations, are referred to as OBSERVATION UNITS.

Table 3. Least Squares treatment means for data in Table 1 – Study S9801.

| | A | B | C | D | E |
|----|---------|-----|------|---------|---------|
| 1 | VARIETY | GID | FERT | YIELD | PHT |
| 2 | B | 100 | 100 | 11.3 | 85 |
| 3 | B | 100 | 200 | 15.95 | 86 |
| 4 | C | 102 | 200 | 12.95 | 93 |
| 5 | C | 102 | 100 | 17.5 | 101.5 |
| 6 | A | 105 | 100 | 14.85 | 85.5 |
| 7 | A | 105 | 200 | 15.4 | 102 |
| 8 | | | | | |
| 9 | SEM | | | 1.72566 | 4.09268 |
| 10 | LSD5% | | | 7.73333 | 18.3408 |

Hence OBSERVATION UNITS are conceptually equivalent to rows in a serially structured spreadsheet, they are the real or conceptual data sources in a study and they are annotated by distinct level combinations of one or more LABELS. Not all LABELS in a study need to be involved in this indexing for every OBSERVATION UNIT. However, STUDY LABELS, with their single levels, are involved in indexing every OBSERVATION UNIT in the Study. Hence OBSERVATION UNITS belong to unique studies. Every study has a STUDY UNIT which is the single observation unit indexed by the level of the STUDY LABEL alone.

DATASETS

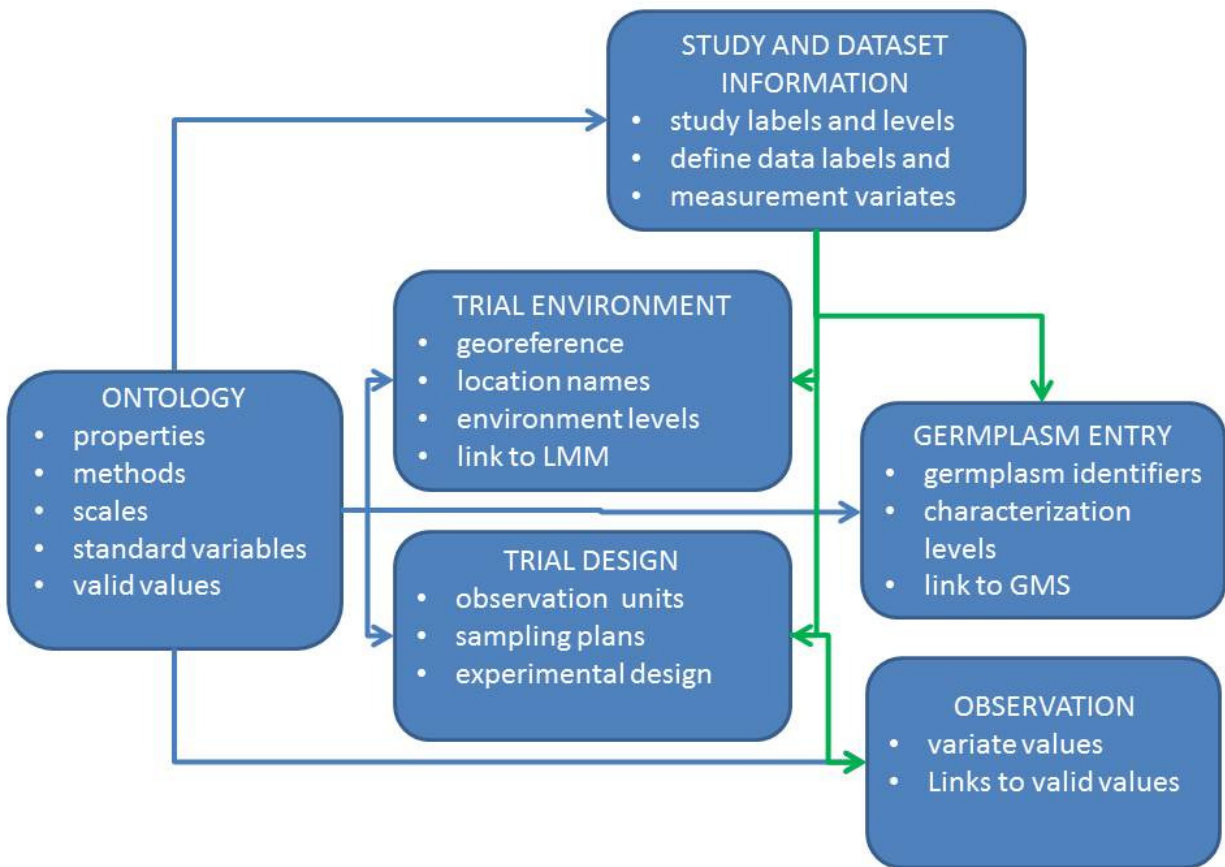
All observation units in a study which have the same labels form a DATASET. We can say that a DATASET is defined by the different level-combinations of a subset of LABELS from the STUDY. The OBSERVATIONS UNITS of the PLOT DATA in Table 1 are indexed by PLOT, REP, MAINPLOT, SUBPLOT, ENTRY, VARIETY, GID and FERT and for the TREATMENT MEANS in Table 2, ENTRY and FERT will define all the OBSERVATION UNITS, but we would like to carry over the other label of ENTRY, VARIETY and GID, as well as all the STUDY LABELS.

LOGICAL DATA MODEL FOR PHENOTYPING DATA FROM FIELD TRIALS

There are five key elements of phenotyping data from field trials which need to be captured in a logical data model. These are:

- The STUDY INFORMATION component records global contextual information about the experiment such as who conducted it, when, why, and who owns the data and models the high level structure of the experiment by describing the datasets that are part of the study for example, data collected about the trial environment(s), data collected on sub-samples, plot data, means and summary data. Each of these datasets is described in terms of the variables that occur in them, which ones are labels giving design and context, and which are variates containing observations made during the experiment. The actual values of these dataset variables are managed by other components of the model depending on their type.
- The TRIAL ENVIRONMENT component manages all data values describing the environments observed in the study including georeference information, place names, growing environments, and overall management practices (non-treatment factors). This component also links to the Location Management Module of the IBDB.
- The GERMPLASM ENTRY component manages all label values describing the germplasm entries in the the experiment including local and global identifiers, names, sources and roles (check or test lines) of the entries. It links to the Germplasm Management System of (GMS) IBDB where unique identification, global nomenclature, ownership and pedigree information is stored.
- The TRIAL DESIGN component manages the treatment and sampling design and structure of the datasets in the study. It enumerates all the observation units in the study and describes their treatment and sampling context in terms of levels of labels describing such features as replication, block, fertilizer treatment etc. the onservation units inherit global information about the study, information about the location and information about the germplasm entries by linkages to the relevant components of the model.
- The OBSERVATION component manages the values of the variates for each dataset.

Figure 1. Key elements of the Logical Data Model for Penotyping Data from Field Trials



These elements are sufficient for managing phenotyping data from any field experiment, however a sixth component is required to facilitate integration of phenotyping data across studies. This is the Ontology Management System (OMS) which identifies comparable elements – labels, variates and values across studies.

BUSINESS RULES FOR USING THE CHADO NATURAL DIVERSITY MODULE TO MANAGE PHENOTYPING DATA FROM FIELD TRIALS

The Ontology Management System

Two ontologies are used to annotate different elements of phenotyping data so that they can be consistently stored in the CHADO ND schema and compatible data can be integrated across different phenotyping studies. Firstly, the IBDB Structure ontology has terms defining properties of different elements of the data model and relationships between them as well as CLASS terms grouping terms of both ontologies into classes. This essentially extends the CHADO ND schema. Secondly, the Crop Ontology defines the variables which describe the context of the phenotyping experiment and which record the observations from the experiments.

The Crop Ontology contains several controlled vocabularies (CV):

- The PROPERTIES CV has terms describing the design and treatment factors applied in phenotyping experiments and the traits being measured in them.
- The METHODS CV has terms describing the protocols by which those properties are applied or measured in phenotyping experiments.
- The SCALES CV has terms describing the scales or units in which the values of the properties are recorded.
- The STANDARD VARIABLES CV has terms defined by combinations of one property term, one method term and one scale term which describe the variables recording the design and observations of phenotyping experiments.
- In addition each categorical variable (one which has a scale constraining its values to a specific set of valid values) has a VALID VALUE CV with terms describing those valid values.

Each of these controlled vocabularies is stored in the *cv* table and the terms from each are in the *cvterm* table. VALID VALUE CVs for categorical variables are named with the *cvterm_id* of the categorical variable and described by the preferred name of that variable.

Figure 2. Controlled vocabularies for Phenotyping Data from Field Trials

| cv_id | name | definition |
|-------|------------|---------------------------------------------------------------------------------------------------|
| 1000 | IBDB TERMS | CV of terms used to annotate relationships and identify objects in the ibdb database |
| 1010 | Properties | Terms defining the factor or trait of a variable |
| 1020 | Methods | Terms representing the methods or protocols by which variables are applied or observed |
| 1030 | Scales | Terms defining the scale or units in which a variable is recorded |
| 1040 | Variables | Standard variables recorded in the dataset defined by a combination of property, method and scale |
| 2010 | 8070 | Study type - assigned (type) |
| 2020 | 8160 | Dataset type - assigned (dtype) |
| 2030 | 8130 | Study release - assigned (date) |
| 2040 | 8135 | Experimental design - assigned (type) |
| 2050 | 8255 | Entry type - assigned (type) |
| 2070 | 8371 | Season - Assigned (Code) |

The IBDB TERMS CV

Terms in the IBDB TERMS ontology belong to one of four classes: IBDB structure, IBDB class, IBDB data type or IBDB element. The relation of belonging to a class is recorded by a record in the cvterm_relationship table with type_id=1225 (is a). Terms in the class IBDB structure describe entities and relationships used in the schema. Terms in IBDB class group terms of an ontology in a hierarchy for easy browsing. Terms in IBDB data type describe the data type of each STANDARD VARIABLE. Terms in IBDB element indicate the element of the logical schema to which a STANDARD VARIABLE belongs, and where its values are stored.

IBDB TERMS which are class terms also have a ‘type’ relationship (1105) to term 1090 ‘Class’

The PROPERTIES CV

Terms in the PROPERTIES CV have an ‘is a’ (1225) relationship to an IBDB class which affords easy browsing.

Table 4. IBDB class terms. Each property and variable ‘is a’ (1225) class member

| cvterm_id | name |
|-----------|----------------------------|
| 1000 | IBDB structure |
| 1002 | IBDB data type |
| 1003 | IBDB element |
| 1001 | IBDB class |
| 1045 | Crop research ontology |
| 1100 | Trial Design |
| 1260 | Seed storage |
| 1270 | Breeding process |
| 1050 | Study condition |
| 1086 | Variate condition |
| 1080 | Trial environment |
| 1280 | Abiotic condition |
| 1300 | Site condition |
| 1310 | Soil condition |
| 1320 | Climatic condition |
| 1290 | Biotic condition |
| 1085 | Trial management |
| 1087 | Germplasm |
| 1321 | Molecular property |
| 1055 | Dataset Condition |
| 1330 | Crop Ontology Trait class |
| 1380 | Passport |
| 1410 | Abiotic stress |
| 1370 | Grain quality |
| 1360 | Biotic stress |
| 1390 | Disease resistance |
| 1400 | Insect and pest resistance |
| 1350 | Morphological |
| 1340 | Agronomic |

1345 Physiological, 1430 Yield components, 1440 Phenology, 1450 Post harvest

The VARIABLES CV

Terms in the VARIABLES CV are related to one PROPERTY term, one METHOD term, and one SCALE term. They also have a type relationship (has type, *type_id*=1105) which defines the data type of its values.

Table 5: Each variable is related to one data type with the ‘has type’ (1105) relationship.

| cvterm_id | name | definition |
|-----------|-------------------------|------------------------------------------------------------------------------------------------------------------|
| 1090 | Class | Class of terms in a cv |
| 1110 | Numeric variable | Variable with numeric values either continuous or integer |
| 1117 | Date variable | Date - numeric value in format yyyyymmdd with least significant parts set to zero according to precision |
| 1118 | Numeric DBID variable | Integer database ID (may be negative) |
| 1120 | Character variable | Variable with character values |
| 1125 | Timestamp variable | Character variable in format yyyy-mm-dd:hh:mm:ss:nnn with least significant parts omitted according to precision |
| 1128 | Character DBID variable | Character database ID |
| 1130 | Categorical variable | Variable with discrete class values (numeric or character all treated as character) |

Each Variable also has a ‘stored in’ relationship (*type_id*=1044) which specifies which component of the schema that stores the values of that variable and where. The list of possible storage elements for any variable is given in Table 6. This means that a STANDARD VARIABLE (combination of PROPERTY, METHOD and SCALE) could appear more than once in the ontology (with a different name) because in one study it might belong to one element of the schema and in another to different element. For example NITROGEN FERTILIZER might be a trial management factor in one study but a treatment factor in another. In the first it belongs to the TRIAL ENVIRONMENT component and in the second it belongs to the TRIAL DESIGN component. This complicates data integration, but is a consequence of splitting the schema into different elements, which for the most part will not overlap. When such overlaps occur, the application layer will have to deal with the data integration.

However the combination of a PROPERTY, METHOD, SCALE and storage ELEMENT is unique and affords integration of all data with these characteristics across all studies in the database. The challenge for the application layer is to have the application know the storage ELEMENT of each variable and we will have to get this from the application templates.

Table 6. Interpretation of ‘stored in or role’ (1044) relationship for terms in the VARIABLE CV

| projectprop.type_id | type of variable | definition and storage location of the values |
|---------------------|---------------------|-----------------------------------------------------------------------------|
| 1010 | Study Information | Study element with values stored in projectprop.value |
| 1011 | Study name | Study name stored in project.name |
| 1012 | Study title | Study title stored in project.description |
| 1015 | Dataset Information | Dataset element with values stored in projectprop.value |
| 1016 | Dataset name | Dataset name stored in project.name |
| 1017 | Dataset description | Dataset description stored in project.description |
| 1020 | Trial environment | Trial environment information stored in nd_geolocationprop.value |
| 1021 | Trial instance | Trial instance number stored in nd_geolocation.description |
| 1022 | Latitude | Georeference data stored in nd_geolocation.latitude |
| 1023 | Longitude | Georeference data stored in nd_geolocation.longitude |
| 1024 | Datum | Georeference geodetic datum stored in nd_geolocation.geodetic_datum |
| 1025 | Altitude | Georeference altitude stored in nd_geolocation.altitude |
| 1030 | Trial design | Field trial design and layout information stored in nd_experimentprop.value |
| 1040 | Germplasm entry | Germplasm entry information stored in stockprop.value |
| 1041 | Entry number | Germplasm entry number unique within in a study stored in stock.uniquename |
| 1042 | Entry GID | GMS germplasm identifier stored in stock.dbxref_id |
| 1046 | Entry designation | GMS germplasm name stored in stock.name |
| 1047 | Entry code | Germplasm entry code assigned within a study stored in stock.value |
| 1043 | Observation variate | Phenotypic data stored in phenotype.value |
| 1048 | Categorical variate | Categorical variate with values stored in phenotype.cvalue_id |

For example term 8250 GREMPLASM IDENTIFIER - ASSIGNED (DBCV) has relationships shown in Table 7.

Table 7: Relationships for a STANDARD VARIABLE term in the CRO

| id | type_id | subject_id | object_id | interpretation of the relationship type | name of the related term |
|------|---------|------------|-----------|-----------------------------------------|--------------------------|
| 2420 | 1225 | 8250 | 1087 | 8250 is a | GERMPLASM TERM |
| 3250 | 1044 | 8250 | 1040 | 8250 stored in | ENTRY ELEMENT |
| 6560 | 1105 | 8250 | 1120 | 8250 has type | CHARACTER VARIABLE |
| 8240 | 1200 | 8250 | 2205 | 8250 has property | GERMPLASM ID |
| 8242 | 1210 | 8250 | 4030 | 8250 has method | ASSIGNED |
| 8244 | 1220 | 8250 | 6000 | 8250 has scale | DBC |

1. Is it necessary to have both ‘has type’ and ‘has scale’ relationships? Don pointed out that the type actually belongs to the scale and I believe we should have that relationship on the scale term. Then the question is whether we should copy the type relationship to the variable term or just go to the scale term to get the type when it is needed? I leave the middleware engineers to decide as they go through the process of developing the middleware. However the question is also valid for the class relationship. This truly belongs to the Property, but has been copied to the variable. Again I leave it to you guys to decide whether that is useful or not. Certainly the fewer relationships we have to manage the better (although they don’t change at all once assigned). A typical scenario would be that a user wishes to browse the variables (not properties) by class. Currently you can get the class directly from the variable term, if we remove the duplication you will have to reach back to the property term to get it. Please advise.

2. Is it necessary to have both ‘is a’ and ‘stored in’ relationships? Here I think it is because the class relationship is used to integrate data of the same property and the ‘stored in’ relationship defines its role in a particular experiment. So you might want a query which asked for all plots which received irrigation. The should have the property ‘Irrigation’ in a Management Practices class, but in some cases it will be a Trial Condition stored in nd_geolocationprop and in other cases it will be a Treatment Factor stored in nd_experimentprop. It may be that in the actual data I have confused these situations, but I think it is for me to straighten that out in the ontology and we should retain both relationships although as mentioned in 1. We may remove the duplicated is a relationship from the variable terms.

The VALID VALUE CVS

Each variable of type 1130 (CATEGORICAL VARIABLE) spawns a VALID VALUE CV as shown in Figure 2. These cvs contain the valid values and their interpretation for the categorical variable. They are named in the *cv* table by the string value of the *cvterm_id* from the VARIABLES CV for the categorical variable to which they belong (although this is not the link which is described below), and they are described in the *cv* table with the description of the categorical variable. For example the valid values for the variable 8135 EXPERIMENTAL DESIGN - ASSIGNED (TYPE) are in cv 8135 as shown in Figure 3.

Figure 3: Valid Values of a categorical variable contained in a sub-cv of the CRO

| cvterm_id | cv_id | name | definition |
|-----------|-------|-----------|----------------------------------------|
| 10100 | 2040 | CRD | COMPLETELY RANDOMIZED DESIGN |
| 10110 | 2040 | RCBD | RANDOMIZED COMPLETE BLOCK DESIGN |
| 10120 | 2040 | ALPHA | ALPHA LATTICE |
| 10130 | 2040 | RIBD | RESOLVABLE INCOMPLETE BLOCK DESIGN |
| 10140 | 2040 | NRIBD | NON RESOLVABLE INCOMPLETE BLOCK DESIGN |
| 10150 | 2040 | NRRCD | NON RESOLVABLE ROW-COLUMN DESIGN |
| 10160 | 2040 | AUGMENTED | AUGMENTED DESIGN |

Categorical variables sometimes require an ordering for their values (either because there is an intrinsic ordering eg Low, Medium, High, or because it makes sense to present them to users (in pick lists for example) in a certain order. The default order is the alphabetical order of *cvterm.name* (with numbers treated in character order). If a different ordering is required each term should have a property in the *cvtermprop* table of *type_id* order (IBDB TERMS *cvterm_id*=1420) with the numerical sequence order for that term as its *value*.

The categorical variable in the VARIABLES CV has a ‘has value’ relationship (*cvterm_id*=1190) to each term in its VALID VALUE CV. For example these relationships are shown in Table 8 for variable 8135.

Table 8: Relationships between a categorical variable and its valid values.

| id | type_id | subject_id | object_id | interpretation of the relationship type | name of the related term |
|-------|---------|------------|-----------|-----------------------------------------|--------------------------|
| 10100 | 1190 | 8135 | 10100 | 8135 HAS VALUE | CRD |
| 10110 | 1190 | 8135 | 10110 | 8136 HAS VALUE | RCBD |
| 10120 | 1190 | 8135 | 10120 | 8137 HAS VALUE | ALPHA |
| 10130 | 1190 | 8135 | 10130 | 8138 HAS VALUE | RIBD |
| 10140 | 1190 | 8135 | 10140 | 8139 HAS VALUE | NRIBD |
| 10150 | 1190 | 8135 | 10150 | 8140 HAS VALUE | NRRCD |
| 10160 | 1190 | 8135 | 10160 | 8141 HAS VALUE | AUGMENTED |



Terms of type 1110 (NUMERIC VARIABLES) may have MINIMUM and MAXIMUM allowable values specified in the *cvtermprop* table as shown in Table 9 for variable SOIL PH.

Table 9: MINIMUM and MAXIMUM allowable values for a numeric variable in the CRO

| cvtermprop_id | cvterm_id | type_id | value | rank |
|---------------|-----------|---------|-------|------|
| 8000 | 8270 | 1113 | 1 | 0 |
| 8010 | 8270 | 1115 | 14 | 0 |

Synonyms and foreign language names and descriptions for terms are stored in the *cvtermsynonym* table.

Figure 4. Synonyms and foreign language names for controlled vocabulary terms

| <input type="checkbox"/> | cvtermsynonym_id | cvterm_id  | synonym | type_id  |
|--------------------------|------------------|---------------------------------------------------------------------------------------------|--------------------|---------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | 1 | 88 | HARVEST | 3012 |
| <input type="checkbox"/> | 2 | 88 | RENDEMENT DE GRAIN | 3013 |
| <input type="checkbox"/> | 5 | 88 | RECOLTE | 3014 |

The Data Management System

STUDY AND DATASET INFORMATION

STUDIES and DATASETS are captured using the CHADO *project* tables. Studies are organized into hierarchical structures of folders. FOLDERS are CHADO projects which can contain sub folders or studies. Studies contain DATASETS. The structure of folders, sub-folders, studies and datasets is managed through the *project_relationship* table. (*type_id* is “is a sub-folder of”, *cvterm_id*=1140, ‘is a study in’ *cvterm_id*=1145 and “is a dataset of”, *cvterm_id*=1150). Folders do not contain information (other than their names and descriptions in the *project* table) but they group sub-folders and studies for easy browsing.

Figure 5. The project and project_relationship tablea

| project_relationship | | | project | | |
|----------------------|--------------------------|------------------------------|------------|--------------------------|------------------------------|
| project_id | name | description | project_id | name | description |
| 1 | TEST STUDIES | Folder for test studies | 1 | TEST STUDIES | Folder for test studies |
| 2 | S9801 | Study 1 of 1998 | 2 | S9801 | Study 1 of 1998 |
| 3 | S9801 - TRIAL CONDITIONS | Trial site data for study 1 | 3 | S9801 - TRIAL CONDITIONS | Trial site data for study 1 |
| 4 | S9801 - PLOT DATA | Plot data for study 1 | 4 | S9801 - PLOT DATA | Plot data for study 1 |
| 5 | S9801 - TREATMENT MEANS | Treatment means for study 1 | 5 | S9801 - TREATMENT MEANS | Treatment means for study 1 |
| 6 | S9801 - VARIETY MEANS | Variety means for study 1 | 6 | S9801 - VARIETY MEANS | Variety means for study 1 |
| 7 | S9801 - FERTILIZER MEANS | Fertilizer means for study 1 | 7 | S9801 - FERTILIZER MEANS | Fertilizer means for study 1 |

| project_relationship_id | subject_project_id | object_project_id | type_id |
|-------------------------|--------------------|-------------------|---------|
| 1 | 2 | 1 | 1145 |
| 2 | 3 | 2 | 1150 |
| 4 | 4 | 2 | 1150 |
| 5 | 5 | 2 | 1150 |
| 6 | 6 | 2 | 1150 |
| 7 | 7 | 2 | 1150 |

Every folder, study and dataset has a unique name and may have a title or description which are stored in the *project.name* and *project.description* fields. If there is a study label with property STUDY, method ASSIGNED and scale DBCV then it contains the name of the study as its single level and it is stored in the *project.name* field. If there is no such label in the study then a unique study name must be supplied by the application.

If there is a study label with property STUDY TITLE, method ASSIGNED and scale TEXT then this is stored in the *project.description* field, otherwise this can be *null*.

If there is a dataset label with property DATASET, method ASSIGNED and scale DBCV then it contains the unique name of the dataset as its single level and it is stored in the *project.name*

field. If there is no such label in the study then a unique dataset name must be supplied by the application for each dataset in the study.

If there is a label with property DATASET TITLE, method ASSIGNED and scale TEXT then this is stored in the *project.description* field of the dataset *project* record, otherwise this can be *null*.

Every study variable in a study is described by three records in the *projectprop* table with *projectprop.project_id* equal to the *project_id* of the study describing the user-supplied name, the user-supplied description of the variable and the *cvterm_id* of the standard variable to which it belongs.

All *projectprop* records for the same label have the same *projectprop.rank* in the *projectprop* table. The rank specifies the user-supplied order of the variables in the study.

The *projectprop.type_id* field indicates whether the *projectprop.value* contains a local name for the variable and if so, where it is stored in the schema, or whether it contains a description (*type_id*=1060) or the link to a standard variable (*type_id*=1070). The interpretation of the *type_id* for the name *projectprop* records is given in Table 6.

As described above the levels of the study name and title labels (if present) are stored in the project table. If the labels are Germplasm Entry labels (eg a study with only one genotype) the values are stored in the *stock* or *stockprop* tables linked to a study observation unit (*nd_experiment* record linked to the study). If the labels are Trial Environment labels (eg a study with only one location) then the levels are stored in the *nd_geolocation* and *nd_geolocationprop* tables linked to a study observation unit (*nd_experiment* record linked to the study).

The levels of all other study labels are stored in a fourth *projectprop* record for each label with the same *projectprop.rank* value as the label description records. If there are any study constant variates (observations or measurements having a single value for the whole study) they are similarly described in the *projectprop* table with *project_id* of the whole study. The values however are stored in the *phenotype* table linked to a study observation unit (*nd_experiment* record linked to the study) .

The values of study variables are cast to all observation units in the study by the study-dataset and dataset-observation unit relationships.

Figure 6. Part of the *projectprop* table for the STYDY (project_id=2)

| projectprop_id | project_id | type_id | value | rank |
|----------------|------------|---------|------------------------|------|
| 1 | 2 | 1011 | STUDY | 1 |
| 2 | 2 | 1070 | 8005 | 1 |
| 3 | 2 | 1060 | STUDY NAME | 1 |
| 4 | 2 | 1012 | TITLE | 2 |
| 5 | 2 | 1070 | 8007 | 2 |
| 6 | 2 | 1060 | TITLE ASSIGNED | 2 |
| 7 | 2 | 1010 | PMKEY | 3 |
| 8 | 2 | 1070 | 8040 | 3 |
| 9 | 2 | 1060 | PROJECT MANAGEMENT KEY | 3 |
| 10 | 2 | 8040 | 123 | 3 |
| 11 | 2 | 1010 | OBJECTIVE | 4 |
| 12 | 2 | 1070 | 8030 | 4 |
| 13 | 2 | 1060 | OBJECTIVE DESCRIBED | 4 |
| 14 | 2 | 8030 | Test the schema | 4 |
| 15 | 2 | 1010 | START DATE | 5 |
| 16 | 2 | 1070 | 8050 | 5 |
| 17 | 2 | 1060 | STUDY START DATE | 5 |
| 18 | 2 | 8050 | 20111202 | 5 |
| 19 | 2 | 1010 | END DATE | 6 |
| 20 | 2 | 1070 | 8060 | 6 |
| 21 | 2 | 1060 | STUDY END DATE | 6 |
| 22 | 2 | 8060 | 20130303 | 6 |

All other variables (labels and variates) in the study belong to one or more datasets and these are described in *projectprop* records, in the same way as study variables, for each dataset in which they occur. As for studies the levels of the dataset name and title labels, if present, are stored in the *project* table. Other labels which are dataset conditions (such as the dataset user id) are stored in the *projectprop* table as with study labels. Values of all other variables in a dataset are stored in tables appropriate to their class, linked to observation units related to the dataset.

Every study and dataset should have user name and user id labels to identify the database user who entered the data for that study or dataset. See the section below on attribution and ownership.

Figure 7. Part of the projectprop table for the plot dataset (*project_id*=4)

| projectprop_id | project_id | type_id | value | rank |
|----------------|------------|---------|------------------|------|
| 63 | 4 | 1021 | TRIAL | 18 |
| 64 | 4 | 1070 | 8170 | 18 |
| 65 | 4 | 1060 | TRIAL NUMBER | 18 |
| 66 | 4 | 1020 | SITE | 19 |
| 67 | 4 | 1070 | 8180 | 19 |
| 68 | 4 | 1060 | TRIAL SITE | 19 |
| 69 | 4 | 1020 | SITE ID | 20 |
| 70 | 4 | 1070 | 8190 | 20 |
| 71 | 4 | 1060 | TRIAL SITE ID | 20 |
| 72 | 4 | 1030 | PLOT | 24 |
| 73 | 4 | 1070 | 8200 | 24 |
| 74 | 4 | 1060 | PLOT NUMBER | 24 |
| 75 | 4 | 1030 | REP | 25 |
| 76 | 4 | 1070 | 8210 | 25 |
| 77 | 4 | 1060 | REPLICATION | 25 |
| 78 | 4 | 1030 | MAINPLOT | 26 |
| 79 | 4 | 1070 | 8220 | 26 |
| 80 | 4 | 1060 | MAIN PLOT NUMBER | 26 |
| 81 | 4 | 1030 | SUBPLOT | 27 |
| 82 | 4 | 1070 | 8200 | 27 |
| 83 | 4 | 1060 | SUB PLOT NUMBER | 27 |
| 84 | 4 | 1040 | ENTRY | 28 |
| 85 | 4 | 1070 | 8230 | 28 |
| 86 | 4 | 1060 | ENTRY NUMBER | 28 |
| ... | | | | |
| 93 | 4 | 1030 | FERT | 31 |
| 94 | 4 | 1070 | 8260 | 31 |
| 95 | 4 | 1060 | FERTILIZER LEVEL | 31 |
| 96 | 4 | 1043 | YIELD | 33 |
| 97 | 4 | 1070 | 18000 | 33 |
| 98 | 4 | 1060 | GRAIN YIELD | 33 |
| 99 | 4 | 1043 | PHT | 34 |
| 100 | 4 | 1070 | 18020 | 34 |
| 101 | 4 | 1060 | PLANT HEIGHT | 34 |
| 102 | 4 | 1048 | BLB | 35 |
| 103 | 4 | 1070 | 18050 | 35 |
| 104 | 4 | 1060 | BLB RESISTANCE | 35 |

The values of labels in datasets are stored in the TRIAL ENVIRONMENT, GERMPLASM ENTRY or TRIAL DESIGN components of the schema and values of the variates of studies or datasets are stored in the OBSERVATION component of the schema. These storage locations are identified from the *projectprop.type_id* of the local name property, which is also reflected in the standard variable cvterm_relationship with type_id=1044 so that the storage location of any variable is known from the dataset side or the standard variable side without linking.

TRIAL ENVIRONMENT

The location/environment component of the logical data model manages information about the trial environment where an experiment is conducted. It uses the *nd_geolocation* table and *nd_geolocationprop* table to store all values of location labels. Information in these tables corresponds to actual physical locations where Field Trials are conducted.

The *nd_geolocation.description* field is used to store the value of factor with property TRIAL INSTANCE, method ENUMERATED and scale NUMBER. If the study comes from the IB Fieldbook this is always present and contains a sequential number 1,2, But studies from other applications may have TRIAL INSTANCE factors with a different scale. If there is no TRIAL INSTANCE factor then simply assign sequential numbers to *nd_geolocation.description* and put all LOCATION labels in the *nd_geolocationprop* table. Georeference properties of the trial site are also stored in the *nd_geolocation* table if available, but all other properties of the trial location (eg site name, side code, site ID and loc_id) are stored in the *nd_geolocationprop* table.

Global information about locations is stored in the Location Management Module of the database, outside the scope of the Phenotyping database, but linked via the location ID label with property LOCATION, method ASSIGNED and scale DBID.

Other labels of the trial site which describe the management or environment of the trial (ie not treatments within the trial) are also stored as properties of the trial environment in the *nd_geolocationprop* table. (eg, irrigation, pesticides, season etc.) In the IB Fieldbook, these are labels of the TRIAL INSTANCE factor.

Figure 8. *nd_geolocation* table and properties SITE ID and SITE for the trial environment stored in the *nd_geolocationprop* table.

| nd_geolocation | | | | | |
|-------------------|-------------|----------|-----------|-------------|----------|
| nd_geolocation_id | description | latitude | longitude | geodetic_da | altitude |
| 1 | 1 | | | | |

| nd_geolocationprop | | | | | |
|-----------------------|-------------------|---------|-----------|------|--|
| nd_geolocationprop_id | nd_geolocation_id | type_id | value | rank | |
| 1 | 1 | 8190 | 10 | 1 | |
| 2 | 1 | 8180 | LOS BANOS | 2 | |

GERMPLASM ENTRIES

The **stock** table maps to the Germplasm component of the logical data model. Entries in the **stock** table represent the Germplasm used in a field trial. Information about the Germplasm is stored in the GMS database, outside the scope of the Phenotyping database.

The **stock.uniquename** field is used to store the label with property GERMPLASM ENTRY, method ENUMERATED and scale NUMBER (ENTRY_NO, **cvterm_id**=8230) this is always present if the study comes from the IBFieldbook, but if it is not present (study from another application) simply store a sequence number for the germplasm entries in the study in this field. (In migration from IBDB V1 the levelno will do).

If there is a label with property GERMPLASM ID, method ASSIGNED and scale DBID (GID, **cvterm_id**=8240) this contains the GID from GMS and they should be stored in the **stock.dbxref_id** field. Else *null*. The **stock** table thus serves as a link to the information in the GMS via the **stock.dbxref_id** field.

If there is a label with property GERMPLASM ID, method ASSIGNED and scale DBCV (DESIGNATION, **cvterm_id**=8250) it contains a germplasm name from the database and its levels should be stored in the **stock.name** field. Else *null*.

If there is another label with property GERMPLASM ID, method ASSIGNED and scale CODE (ENTRY_CODE, **cvterm_id**=8300) it contains a study level entry code for the germplasm and its levels should be stored in the **stock.value** field and its **cvterm_id** in the **stock.type_id** field. Else both are *null*.

So what to use the type_id for since it is obligatory? All the variables have fixed and known cvterms so what other data could be useful? Put 8300 (ENTRY_CODE) for now.

Figure 9: Information on the entries in the experiment in the **stock** table.

| stock | | | | | | | | |
|----------|-----------|-------------|------|------------|-------|-------------|---------|-------------|
| stock_id | dbxref_id | organism_id | name | uniquename | value | description | type_id | is_obsolete |
| 1 | 100 | | B | 2 | | | 8300 | 0 |
| 2 | 102 | | C | 3 | | | 8300 | 0 |
| 3 | 105 | | A | 1 | | | 8300 | 0 |

The **stockprop** table contains levels of any other labels with relationship ‘stored in’ (**cvterm_id**=1044) pointing to ‘Germplasm entry’ (**cvterm_id**=1040) for example other names or whether a certain seed is a control/check in a field trial, or its seed source - where you got the seed or a label for the pedigree.

TRIAL DESIGN

The Field Trial component of the Logical Data Model uses the **nd_experiment** tables. The information stored in the **nd_experiment** and supporting tables describes the design of the nursery or field trial and/or the structure of derived data coming from summary processes or statistical analysis. General information about the study which is contained in study labels is

inherited by all the observation units of all datasets belonging to the study. Information on the germplasm included in the study is indicated by links to the *stock* table, and information on environments used in the study is provided by links to the *nd_geolocation* table.

If a factor describes a management practice which is a treatment in the experiment eg fertilization or irrigation, it is part of the Field Trial design, but if it applies to the whole environment or trial site, for example, irrigation is sometimes applied to an entire location, then it is managed in the Trial Environment component. In other cases irrigation varies by plot in which case it will be managed as part of the Field Trial component with levels stored for each plot in the *nd_experimentprop* table..

Each record in the *nd_experiment* table corresponds to one observation unit in a dataset and serves to link a specific combination of label values (levels) to a specific set of variate values. The levels of the design labels (treatments, design and lay-out) specify the context of the experiment or observation unit on which the associated values of the variates were observed. The study labels, environment labels and entry labels are associated by links from those components to the *nd_experiment* records (observation units).

Figure 10. Observation units of the plot data listed in the *nd_experiment* table.

| nd_experiment | | |
|---------------|-------------|---------|
| nd_experim | nd_geolocat | type_id |
| 1 | 1 | 1010 |
| 2 | 1 | 1020 |
| 3 | 1 | 1155 |
| 4 | 1 | 1155 |
| 5 | 1 | 1155 |
| 6 | 1 | 1155 |
| 7 | 1 | 1155 |
| 8 | 1 | 1155 |
| 9 | 1 | 1155 |
| 10 | 1 | 1155 |
| 11 | 1 | 1155 |
| 12 | 1 | 1155 |
| 13 | 1 | 1155 |
| 14 | 1 | 1155 |
| 15 | 1 | 1170 |
| 16 | 1 | 1170 |
| 17 | 1 | 1170 |

Each observation unit has a type indicated by ***nd_experiment.type_id*** linking to ***cvterm***. Some types are:

- a) Type = Study (***cvterm_id***=1010) linking levels of labels which apply to the whole study (like the PI's name) to observations which might have been made at the whole study level – like water condition eg irrigated or not.
- b) Type = Dataset (***cvterm_id***=1015) linking levels of the whole dataset environment to observations made at this level.
- c) Type = Trial Environment (***cvterm_id***=1020) linking levels of labels which apply to each instance of a trial (ie each environment where a trail is repeated). Usage of properties is similar to study above.
- d) Type =Field Plot (***cvterm_id***=1155). A field trial. Linked by ***proj_id*** (to what?) and links into the stock table. The properties describe the plot. Raw observations are linked from here in the phenotype table.
- e) Type=Sample. (***cvterm_id***=1160) For a sample unit smaller than a plot
- f) Type = Average (***cvterm_id***=1170). Linked by ***proj_id***, links to the stock table, properties determined by what is being tracked. Means are linked from here in the phenotype table.
- g) Type = Summary (***cvterm_id***=1180) for summary statistics like SEs and LSDs. Linked by ***proj_id***, links to the stock table, properties similar to plot.

Each observation unit (***nd_experiment*** record) belongs to a project (study or dataset) and this is recorded in the ***nd_experiment_project*** table which allows many to many linkages between ***nd_experiment*** records and projects although in the breeding context each observation unit will probably belong to one dataset

Each ***nd_experiment*** record links to Trial Environment and Location labels via the ***nd_experiment.nd_geolocation_id*** field. If the location/environment information is not available or is not relevant (eg for the mean over several locations) then this field is set to 1 – a ‘not specified’ environment (since it is not allowed to be *null*).

Details of the germplasm applied to each ***nd_experiment*** record are linked via the ***nd_experiment_stock*** linkage table which allows many to many linkages between ***nd_experiment*** records and stock records although in the breeding context this will almost always be one stock to many observation units (***nd_experiment*** records). This table requires a ***type_id*** for each link. It is not clear what information that should carry so we can set it to 1000 for now.

Levels of other labels describing the experiment context are supplied in the ***nd_experimentprop*** table and these are linked to their standard variable ID via the ***nd_experimentprop.type_id***. The ***nd_experimentprop.rank*** is only needed if you have two values of the same ***type_id*** for the same ***nd_experiment_id*** so it can be left 0 for the moment.

Figure 11. Design and treatment levels for plot units in the *nd_experimentprop* table.

| nd_experimentprop_id | nd_experiment_id | type_id | value | rank |
|----------------------|------------------|---------|-------|------|
| 1 | 3 | 8200 | 1 | 1 |
| 2 | 3 | 8210 | 1 | 2 |
| 3 | 3 | 8221 | 1 | 3 |
| 4 | 3 | 8222 | 1 | 4 |
| 5 | 3 | 8260 | 100 | 7 |
| 6 | 4 | 8200 | 2 | 1 |
| 7 | 4 | 8210 | 1 | 2 |
| 8 | 4 | 8221 | 1 | 3 |
| 9 | 4 | 8222 | 2 | 4 |
| 10 | 4 | 8260 | 200 | 7 |
| 11 | 5 | 8200 | 3 | 1 |
| 12 | 5 | 8210 | 1 | 2 |
| 13 | 5 | 8221 | 2 | 3 |
| 14 | 5 | 8222 | 1 | 4 |
| 15 | 5 | 8260 | 200 | 7 |
| 16 | 6 | 8200 | 4 | 1 |
| 17 | 6 | 8210 | 1 | 2 |
| 18 | 6 | 8221 | 2 | 3 |
| 19 | 6 | 8222 | 2 | 4 |
| 20 | 6 | 8260 | 100 | 7 |

...

| | | | | |
|----|----|------|-----|---|
| 56 | 14 | 8200 | 12 | 1 |
| 57 | 14 | 8260 | 100 | 1 |
| 58 | 14 | 8210 | 2 | 2 |
| 59 | 14 | 8221 | 3 | 3 |
| 60 | 14 | 8222 | 2 | 4 |
| 61 | 15 | 8260 | 100 | 1 |
| 62 | 16 | 8260 | 200 | 1 |
| 63 | 17 | 8260 | 200 | 1 |
| 64 | 18 | 8260 | 100 | 1 |
| 65 | 19 | 8260 | 100 | 1 |
| 66 | 20 | 8260 | 200 | 1 |
| 67 | 26 | 8260 | 100 | 1 |
| 68 | 27 | 8260 | 200 | 1 |

OBSERVATION

Variate values are recorded in the *phenotype* table. All phenotype records are linked to the appropriate *nd_experiment* record via the *nd_experiment_phenotype* linkage table. This allows many to many linkages between *nd_experiment* records and phenotype records although in breeding trials it will always be one *nd_experiment* record to one or more phenotype records (possibly none). The type of the observation unit (*nd_experiment* record) indicates how the values are obtained. In the case of experiments of type Plot the observations in the phenotype table will be the directly observed raw data about plants in field trial (e.g. yield, plant height). In the case of experiments of type Average or Summary, the observations in the phenotype table will be the results of statistical analysis on the raw data from Plot experiments. These results will be stored rather than computed in the database due to the complexity of the analysis even for simple means calculations.

The variate value is stored in the *phenotype.value* field whether it is numeric or character except for categorical variates for which the valid values are stored in the *cvterm* table and the actual value is indicated by a link from *phenotype.cvlaue_id*.

The *phenotype.observable_id* field links the phenotype value to the standard variable (STDVAR) cv term in the *cvterm* table. The *phenotype.name* field also stores the cvterm id for the standard variable so that links can easily be made to the standard variable id in the *projectprop* table when browsing datasets.

Figure 12. Phenotype values for plot data in the *phenotype* table.

| phenotype | | | | | | | |
|--------------|------------|-------|---------------|---------|-------|-----------|----------|
| phenotype_id | uniquename | name | observable_id | attr_id | value | cvalue_id | assay_id |
| 1 | 1 | 18000 | 18000 | | 10.3 | | |
| 2 | 2 | 18020 | 18020 | | 80 | | |
| 3 | 3 | 18050 | 18050 | | 3 | 19030 | |
| 4 | 4 | 18000 | 18000 | | 12.7 | | |
| 5 | 5 | 18020 | 18020 | | 85 | | |
| 6 | 6 | 18050 | 18050 | | 3 | 19030 | |
| 7 | 7 | 18000 | 18000 | | 18.7 | | |
| 8 | 8 | 18020 | 18020 | | 103 | | |
| 9 | 9 | 18050 | 18050 | | 5 | 19050 | |
| 10 | 10 | 18000 | 18000 | | 13.7 | | |
| 11 | 11 | 18020 | 18020 | | 88 | | |
| 12 | 12 | 18050 | 18050 | | 4 | 19040 | |
| 13 | 13 | 18000 | 18000 | | 12.6 | | |
| 14 | 14 | 18020 | 18020 | | 79 | | |
| 15 | 15 | 18050 | 18050 | | 2 | 19020 | |
| 16 | 16 | 18000 | 18000 | | 16.7 | | |
| 17 | 17 | 18020 | 18020 | | 102 | | |
| 18 | 18 | 18050 | 18050 | | 1 | 19010 | |

OWNERSHIP AND ATTRIBUTION

Every data value can be traced back to a database user. To do this applications must add a UID variable at the appropriate level for the data value. Every study must have a Study_UID (*cvterm_id*=8020) variable, if a different user adds a dataset to the study, it must have a Dataset_UID variable in the dataset with its value there also, if a different user adds a record to a dataset it must have an OU_UID in the *nd_experimentprop* table. If a different user adds a variable to a dataset it must have a Variable_UID property attached to the variable description in the dataset and if a different user adds or changes a phenotype value .. we need a phenotypeprop table! Value_UID

Any value then 'belongs' to its closest UID value in the order phenotype, variable, observation unit, dataset and study.

The same system could be used to timestamp data and different levels of precision.

DIFFERENCES BETWEEN IBDB V1 AND IBDB V2

The key difference between the two versions is that the original concept of user defined factors that was central to IBDB V1 has been dropped. Factors were subsets of the labels in a study, one of which (the factor label) was required to be discriminate between all levels of the property concerned. Other labels in the factor could have one to many relationships with these levels. In IBDB V2 labels are divided into components of the data model – study, trial environment, germplasm entry, trial design. In effect each study has these four factors.

The old constraint that the factor label had to be maximally discriminatory has been implemented slightly differently by assuming or imposing an ID in each factor – study id, trial instance, germplasm entry, and field plot. These are just numbers and may or may not be explicitly included in the component labels for a particular study. If they are not included the model simply inserts them into *project.project_id*, *nd_geolocation_id* and *description*, *stock.uniquename* and *nd_experiment_id*.

Another difference is that the values of variables are stored in their respective components instead of just splitting them into label levels and data values. Also these are no longer split as numeric and character. All numeric values are stored as character strings. However there is now a split between categorical and other types of variables with values of categorical variables being stored as the ids (*cvterm_ids*) of their valid values from the ontology management system.

Another thing that has changed is the concept of effect and representation. These are now rolled into the concept of dataset and managed through the *project_relationship* table.