IVS WG-4:
VLBI Data Structures

John Gipson
NVI,Inc/NASA GSFC

7th IVS General Meeting
06-March-2012
Madrid, Spain

Members

Chair	John Gipson
Analysis Coordinator	Axel Nothnagel
Haystack/Correlator Representative	Roger Cappalo
GSFC/Calc/Solve	David Gordon Dan MacMillan
IAA/QUASAR	Sergey Kurbodov Elena Skurhina
JPL/Modest	Chris Jacobs
Occam	Oleg Titov
Vienna	Johannes Boehm
Steelbreeze Formally MAO, now at GSFC	Sergei Bolotin
NICT	Thomas Hobiger Hiroshi Takiguchi

Inspired by work of Leonid Petrov and Anne-Marie Gontier.

Working Group Charter

The Working Group will ***examine the data structure currently used*** in VLBI data processing and ***investigate what data structure is likely to be needed in the future***.

It will ***design a data structure that meets current and anticipated requirements*** for individual VLBI sessions including a ***cataloging, archiving and distribution*** system.

Further, it will ***prepare the transition capability*** through ***conversion of the current data structure*** as well as ***cataloging and archiving software*** to the new system.

Good News and Bad News

Good News:

- Format and structure well defined
- All geodetic VLBI databases converted to the new format
- Currently can serve as replacement for ‘superfiles’
- Publicly available (in beta form) at:

htpp://gemini.gsfc.nasa.gov/pub/openDB

Good News and Bad News

Good News:

- Format and structure well defined
- All geodetic VLBI databases converted to the new format
- Currently can serve as replacement for ‘superfiles’
- Publicly available (in beta form) at:

htpp://gemini.gsfc.nasa.gov/public/OpenDB

Bad News--still work to be done.

- Need to interface to C5++, Occam
- Iron out data archiving, submission protocal
- Develop catalog system
- Use OpenDB format in all stages of calc/solve processing

Mark3 Databases.

30+ years old. Used to archive and transmit IVS sessions.

A product of its time:

- Designed to run on systems with 20k (!!) memory
- Designed before Fortran had strings

Furthermore…

- Custom format.
- Difficult to port
- Slow. Leads to two formats used in calc/solve.
  - Databases archive information.
  - Superfiles used in analysis.
- Baseline oriented
- Tremendous redundancy of some kinds of data.

Mark3 Databases.

- A database for each band, even though much data redundant.
- Deeply tied up with calc/solve processing
  - Contains lots of information of limited or no use to broader community

- Mixture of data types
  - Observational
  - Theoretical
  - Solution set-up

Limited user community (20 users?)

- Few people to contribute to improvements.

Mark3 Databases.

Mark3 databases are both:

1. A ways of storing data.
   - Ø. Data is stored in a custom binary format.
   - Ø. Data is self-describing
   - Ø. Data is accessed via calls to a proprietary database handler.

2. A way of organizing data.
   - Ø. Data is organized by “Lcodes”.
   - Ø. Type 1 Lcodes are things that describe the session as a whole. Stations, sources, positions, clock breaks, constraints…
   - Ø. Type 2 &3 Lcodes are data related to a particular observation. Group delay, pointing, ambiguity, ionosphere, editing.

Design Goals

Absolute requirement:

Handle current and anticipated VLBI data needs

Low level goals:

1. Reduce redundancy
2. Ease of access.
3. Speed of access.
4. Different platforms, different languages

Design Goals

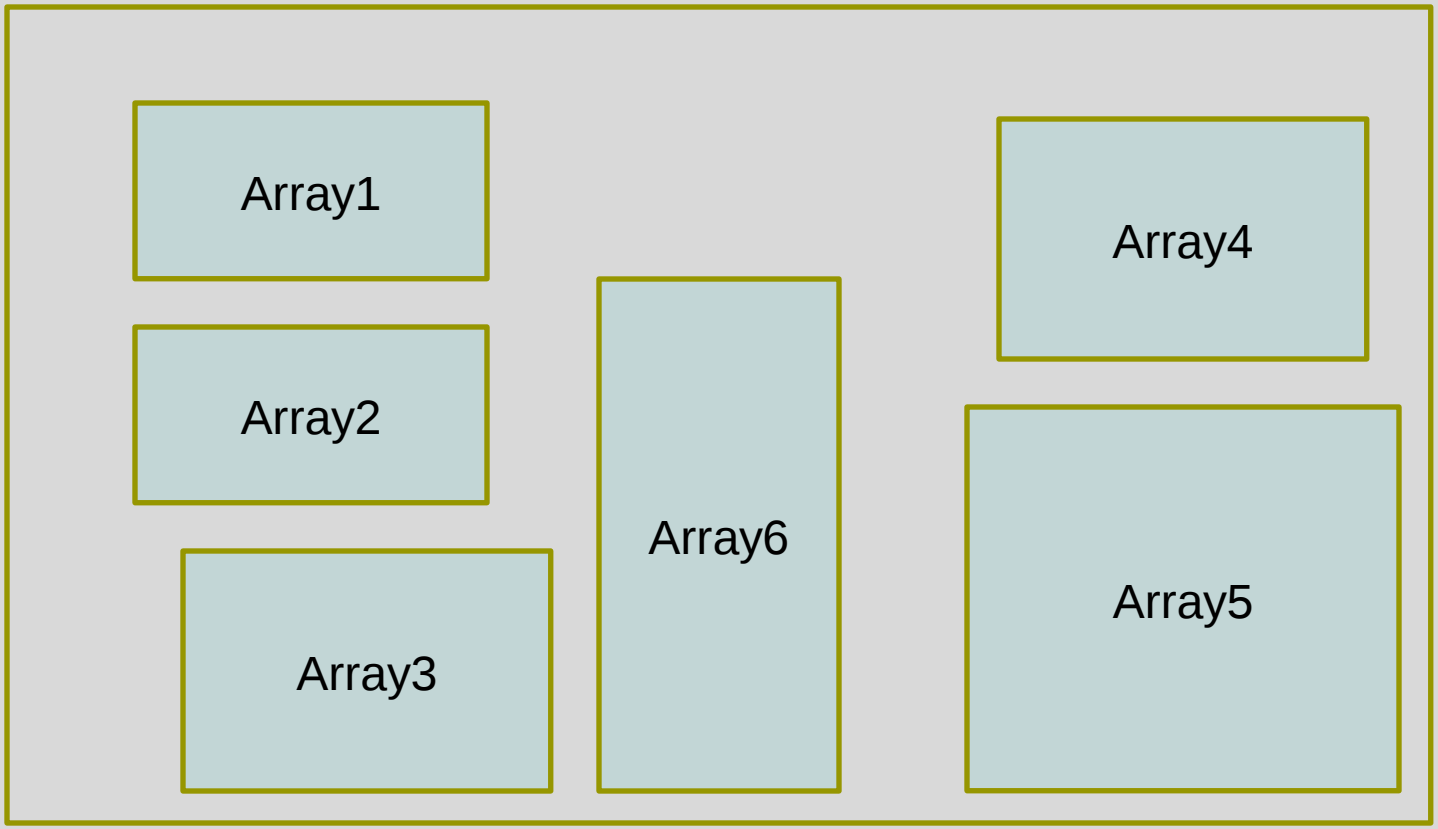
High level goals:

1. Flexibility
2. Easy interchange of data.
3. Separation of “observations” from “models” and “theory”
4. Ability to easily access most common parts of the data
5. Ability to access data at different levels of abstraction.
6. Completeness

Design Goals, Take Two

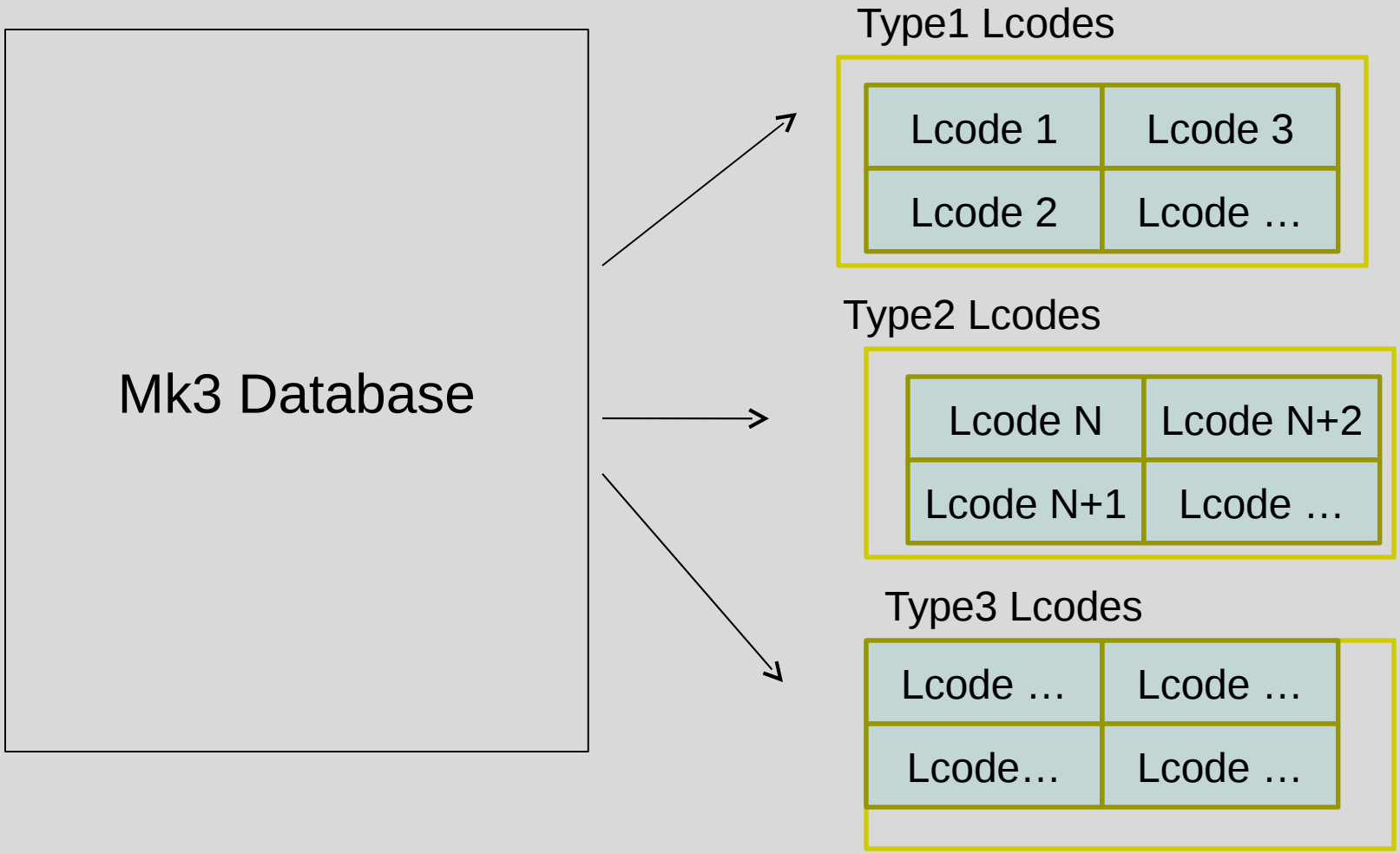
Goal	Format	Organization	Done?
Low Level Goals			
Reduce Redundancy		□	
Ease of Access	□		
Speed of Access	□		
Many Languages, Platforms	□		
High Level Goals			
Flexibility		□	
Easy interchange of sub-sets of the data.	□	□	
Separate observables, models, theoreticals		□	
Separate things that change from things that don't		□	
Easy access to commonly used parts of the data.		□	
Data at different levels of abstraction.		□	
Completeness		□	

NetCDF Files



A NetCDF file can contain an arbitrary number of arrays.
The arrays can differ in dimensions and type (byte, short, integer, real, double).
The arrays can have attributes like name, unit, long-name, description associated with them.

Hobiger's NetCDF databases



A Mark3 database is split into 3 NetCDF files.
There is a 1-1 correspondence between Lcodes and Net CDF arrays

John Gipson NVI, Inc./NASA GSFC
IVS Working Group IV

Design Goals, Take Two

Goal	Format	Organization	Done?
Low Level Goals			
Reduce Redundancy		✓	
Ease of Access	✓		✓
Speed of Access	✓		✓
Many Languages, Platforms	✓		✓
High Level Goals			
Flexibility		✓	?
Easy interchange of sub-sets of the data.	✓	✓	
Separate observables, models, theoreticals		✓	
Separate things that change from things that don't		✓	
Easy access to commonly used parts of the data.		✓	
Data at different levels of abstraction.		✓	
Completeness		✓	

Characterizing VLBI Session Data

Various ways of characterizing data…

1. Scope: Session, Station, Scan or Observation.
2. Origin: Calculated (theoretical delay, ocean loading) or measured (cable cal, met data), or produced by a program (fringe).
3. Frequency of change. Observables never change. Editing criteria might.
4. Raw data vs processed data.
5. Commonly used vs rarely used.

Keep Similar Data Together

Proposal: Gather data that is similar in scope, origin, physical effect, frequency of change. Store in its own file.

1. Experiment info: everything known about experiment beforehand.
2. Met data for a particular station
3. Calibrations
4. Physical and geophysical effects calculable beforehand: relativity, tidal ocean loading, etc.
5. Physical and geophysical effects calculable afterwards: atmosphere loading, hydrological loading, etc.
6. Observables and observation related data.
7. Editing and Ambiguity
8. Information about the solution—clock breaks, constraints, reweighting constants.
9. Less commonly used observation related data

Advantages of This Split

1. Easy to add new data types.
2. Easy testing of new models.
3. Items that are not expected to change are separated from items that may change.
4. Data is separated from models.
5. Lends itself to building up the session piece by piece.
6. We delay discussion of what the VLBI2010 observable format should look like.
7. Commonly used data separated from less commonly used data.
8. You can download only that part of the data you are interested in —“NGS cards” or whole database.

Organizing the Data

Now that we have split the data, how do we gather it up?

We wrap it up using ***wrappers.***

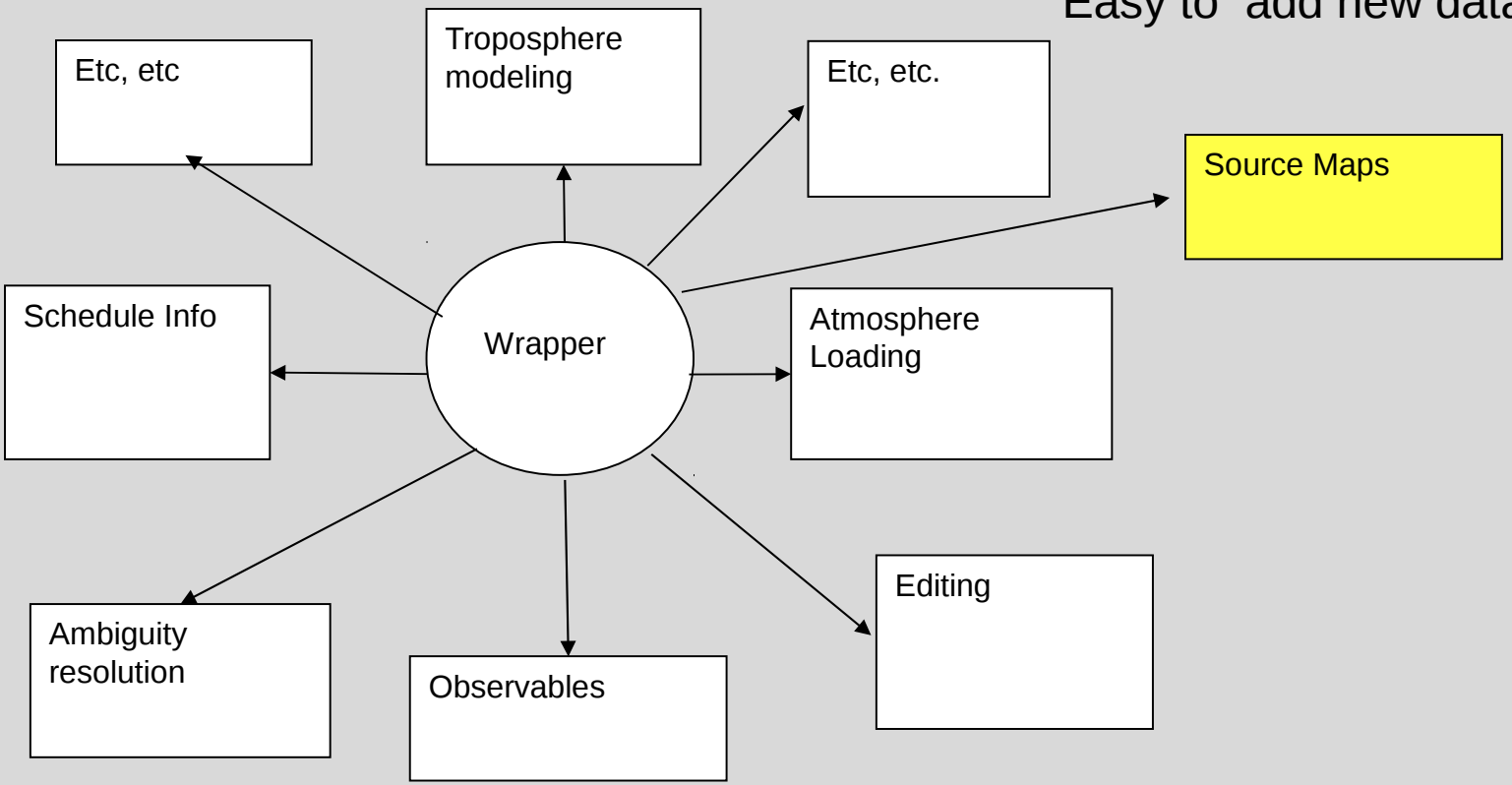
A wrapper is an ASCII file that contains pointers to files that contain data about a session.

A pointer is an instruction about where to find the data.

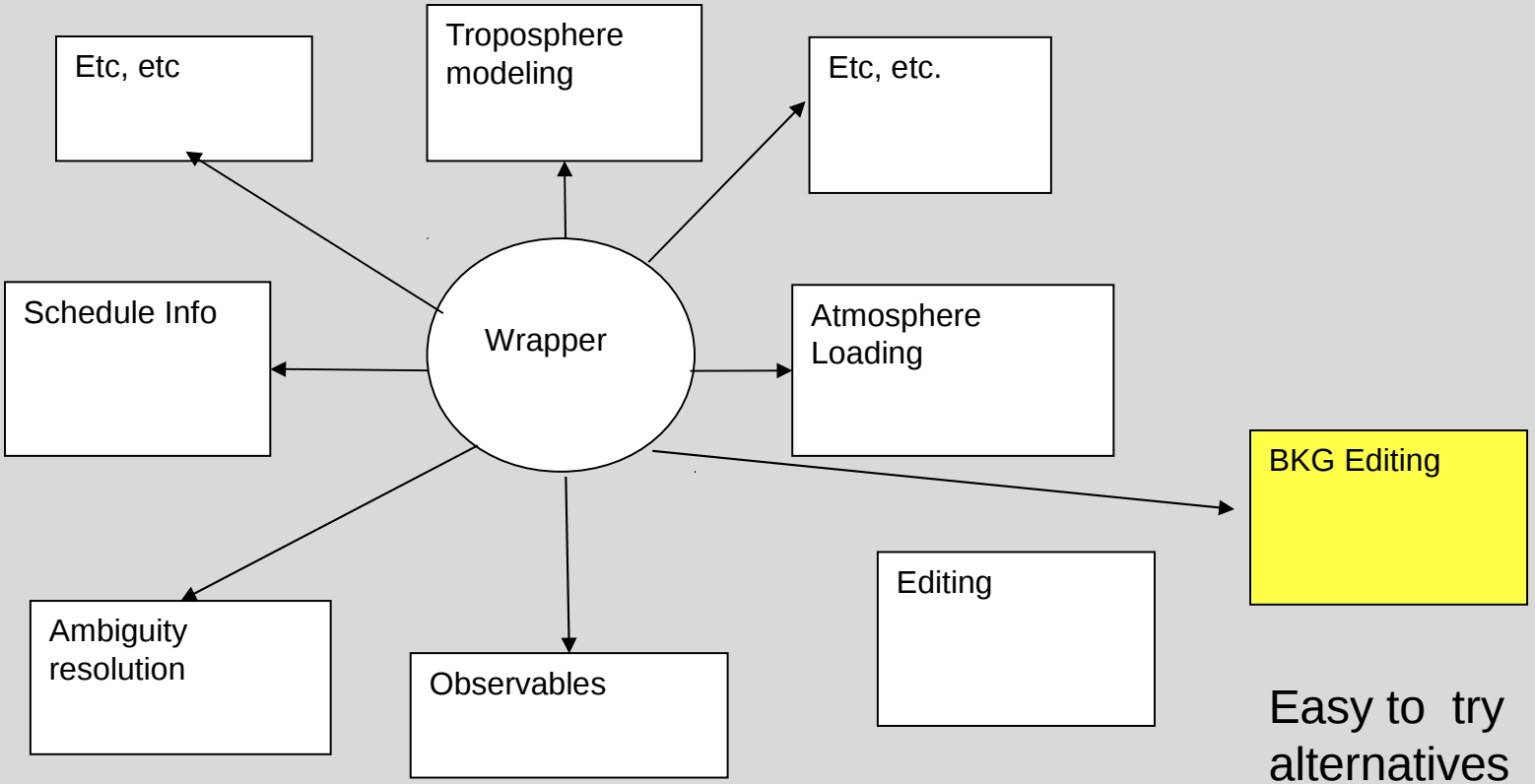
Simplest case is a location on the disk.

Wrappers Organize Session Data

Easy to add new data types



Wrappers Organize Session Data



Easy to try alternatives

Advantages of Wrappers

Easily adapts to new data types (of course you must modify the analysis software).

Can easily swap in and out ‘plug compatible’ files.

Can have private wrappers for R&D.

As models change, don’t have to replace entire database—only need to update appropriate file and wrapper.

Inside a Wrapper

History

Description

Session Data

Kokee Dependent Data

Observation Data

```
Begin History
CreateTimeTag 2009Jun20-12:22:22
Createdby  JohnGipson
End History
! This is a comment.
Begin Description
This is a simple wrapper file for the data in NGS cards.
End Description
! -----another comment.
Begin Session
Session I1234
Head.nc
End Session
! ***start the station sections.
Begin Station KOKEE
! KOKEE must be one of the station names in Head.nc
Default_dir KOKEE
AzEl.nc
Met.nc
Cal_kCable.nc
End Station KOKEE
…. OMIT WETTZELL for brevity
! **** Start the observation section
Begin Observation
Default_Dir Obs
ObsIndex.nc
Obs_bX.nc
Obs_bS.nc
Default_Dir ObsEdit
Edit_bX.nc
Ambig_bX.nc
Ambig_bS.nc
Iono_bX.nc
End Observation
```

Design Goals, Take Three

Goal	Format	Organization	Done?
Low Level Goals			
Reduce Redundancy		▯	▯
Ease of Access	▯		▯
Speed of Access	▯		▯
Many Languages, Platforms	▯		▯
High Level Goals			
Flexibility		▯	▯
Easy interchange of sub-sets of the data.	▯	▯	▯
Separate observables, models, theoreticals		▯	▯
Separate things that change from things that don't		▯	▯
Easy access to commonly used parts of the data.		▯	▯
Data at different levels of abstraction.		▯	▯
Completeness		▯	▯

What do we call this scheme?

openDB format

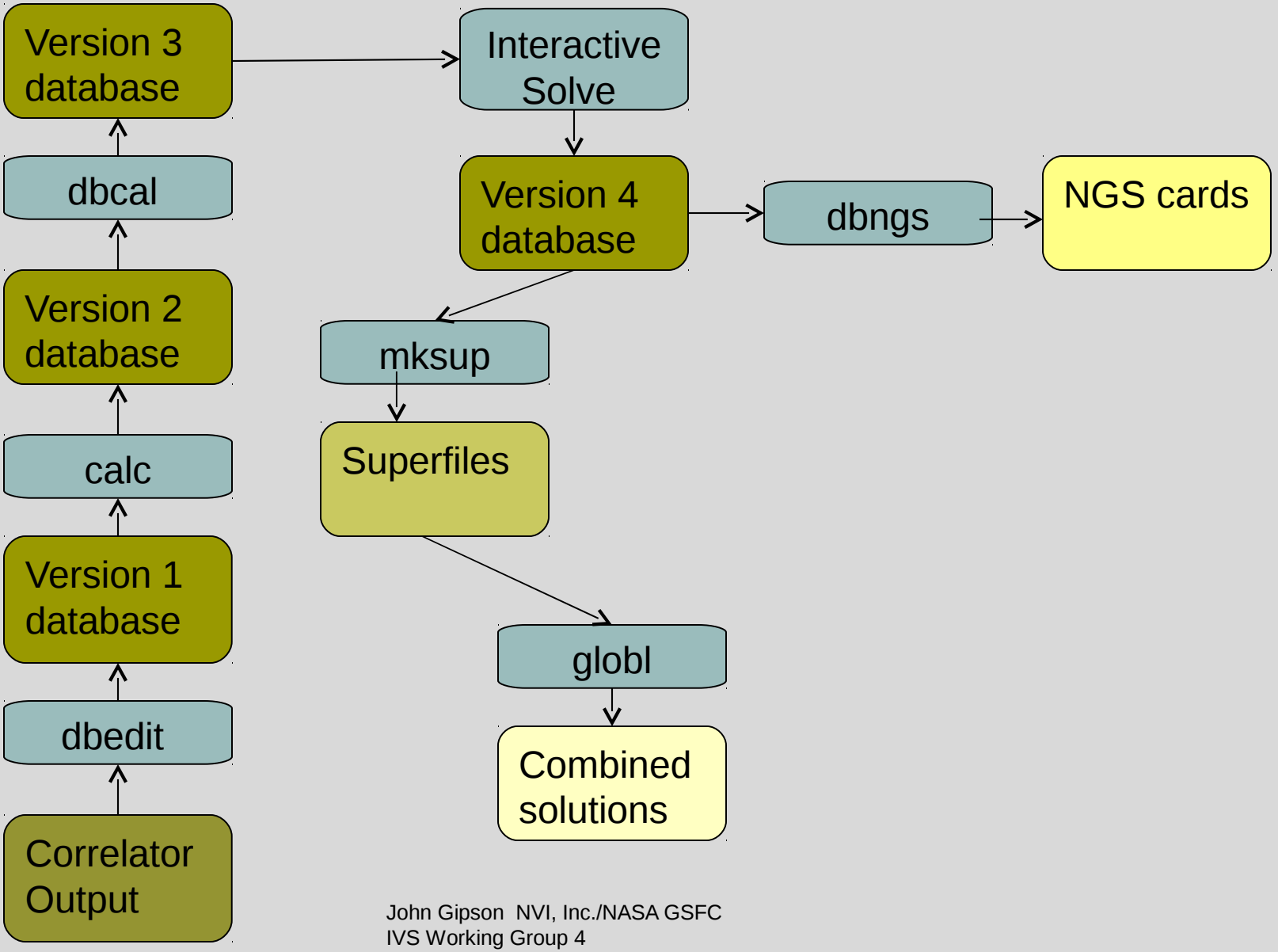
Acknowledges long history of Mark3 databases.

Emphasizes open structure of new format.

Transition to openDB

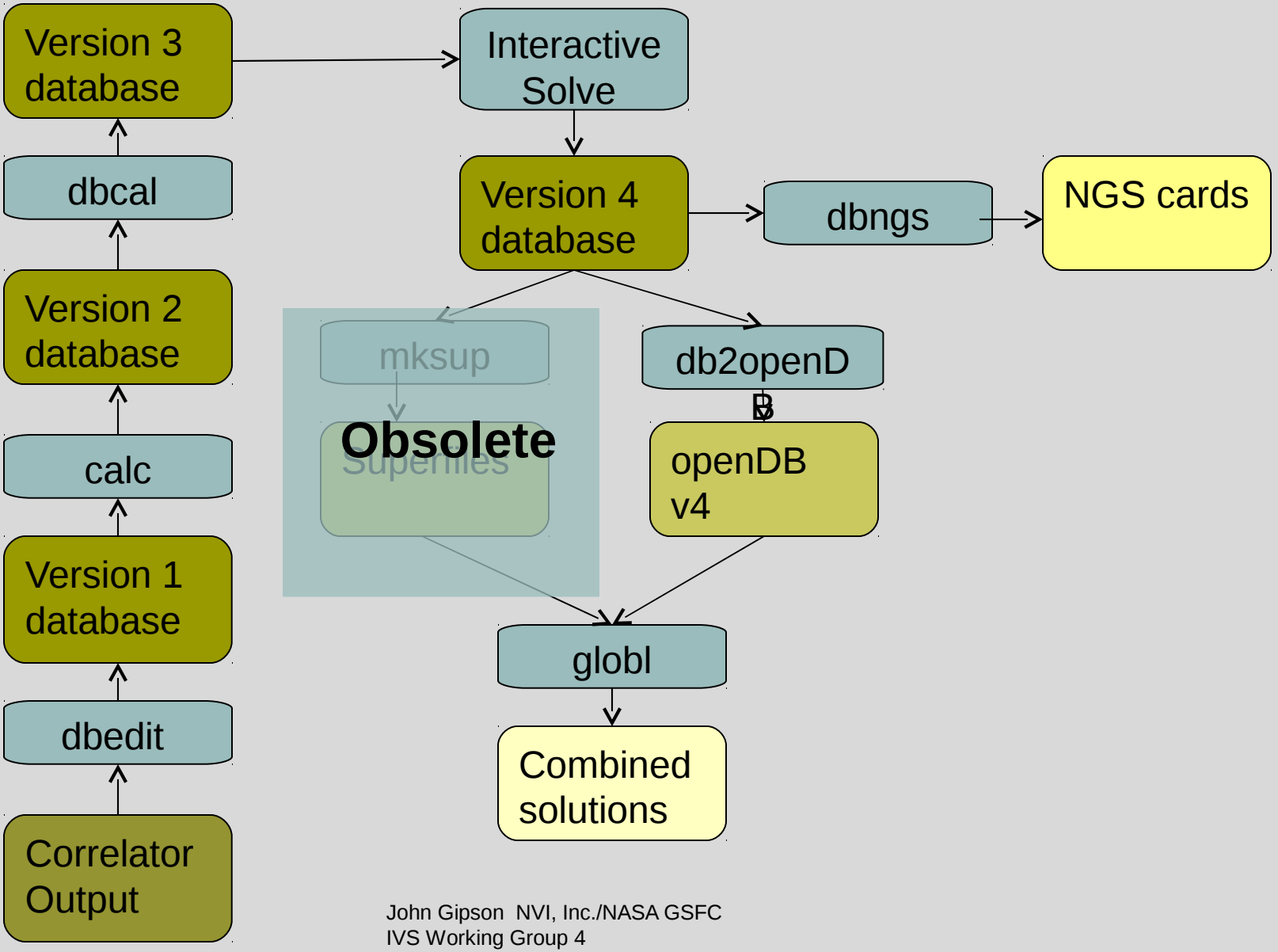
1. Conversion of Mark3 databases to new format.
   A. Wrote utility to convert “NGS-card” subset to OpenDB July 2009.
   B. Started with software that T. Hobiger provided.
2. Steelbreeze
   A. Partial conversion Sep 2009. Uses NetCDF as storage format.
   B. Timing penalty of 40 microseconds/obs. No optimization.
      C. 6 million obs * 40 us=240 seconds penalty=6 minutes.
3. VieVs . (Vienna VLBI Group). Matthias Madzak.
   A. Modified to use new format: March-April 2010.
   B. Introduced new file type—trp.nc. Contains information about troposphere modeling.
4. Occam
   A. Began in fall in 2011
5. C5++
   A. Will begin shortly.

Calc/Solve Transition



John Gipson NVI, Inc./NASA GSFC
IVS Working Group 4

Calc/Solve Transition



John Gipson NVI, Inc./NASA GSFC
IVS Working Group 4

Calc/Solve Transition

Transition from superfiles to openDB began Fall 2010.

Essentially completed Spring 2012.

Handfull (~5) of databases that there are still issues with. (What to do if lcodes are missing or corrupt?)

Processing time:

Large databases (Conts, RDVs) 15-20% faster with openDB than superfiles.

R1s & R4s about the same.

On average about openDB about 6% slower.

Userpartial of solve does not *yet* work.

Calc/Solve Transition

Need to work up the processing tree.

Dbcal

Calc

Dbedit.

Need to develop catalog system

Other Issues/More Work

View current openDB format as ‘beta-version’.

Expect overall structure to remain intact, but some fine-tuning.

Things still to be done:

Protocol for data-transfer and dissemination within IVS.

Catalog for IVS data.

Protocol for adding new data types.

JPL interface

Questions & Comments

?