Data and Data Models for the TVO

Claudio Gheller

InterUniversitary Computing Center CINECA, Bologna

Ugo Becciani

Astrophysical Observatory of Catania

Fabio Pasian, Riccardo Smareglia

Astronomical Observatory of Trieste - INAF

Cambridge, February 2006

Theoretical data: introduction

The main purpose of the ITVO is to create a distributed database of simulated data accessible from anywhere in a easy and transparent way, and to include some services to allow the user to download data extract information from them

The infrastructural details will be introduced by R. Smareglia in a different talk.

My focus is on data model in the TVO framework

Our reference works are:

- The data model for observational data (see IVOA web site)
- The paper by L.D. Shaw, N.A. Walton, and J.P. Ostriker June 2004, "*Towards the Incorporation of Cosmological Simulation Data into the Virtual Observatory*" (referred as pub1)

Theoretical data: produced vs. derived

Produced data are the output of a simulation or a simple combination of them. They corresponds to observational images.

Derived data are the result of (more or less) complex elaboration of Produced data. They corresponds to catalogues/surveys

HERE WE FOCUS ON PRODUCED DATA

For numerical simulations, we can identify two classes of data associated to a specific realization. The first is represented by Raw Data, which are the direct outcome of the simulation. They are usually large binary files which correspond to the images in observations. For this kind of data the crucial issue is *performance*, in any kind of operation (download, cut, trim…) that one can think to perform on it.
The second class is composed by data that are calculated from the bulk of raw data. For example surveys of reconstructed galaxies or galaxy clusters (which corresponds to the analogous observational catalogues) or maps of various physical observable (X-ray, radio, gamma, SZ effect…). This data are much closer to observational data .

Numerical simulations: case study

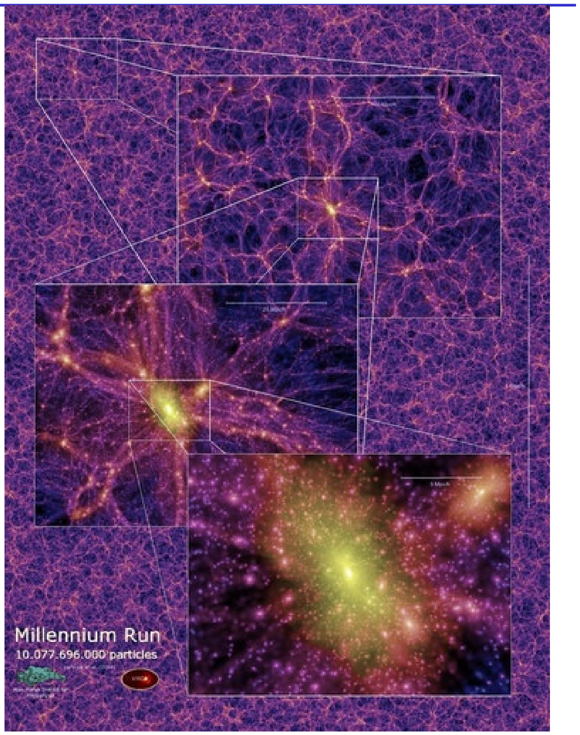
In order to analyze the needs of data produced by numerical simulations, we have considered a wide spectrum of applications:

- Particle based Cosmological simulations
- Grid based Cosmological simulations
- Magnatohydrodynamics simulations
- Planck mission simulated data
- ...

(thanks to V. Antonuccio, G. Bodo, S. Borgani, N. Lanza, L. Tornatore)

Numerical simulations data. Examples :

1. The GADGET code



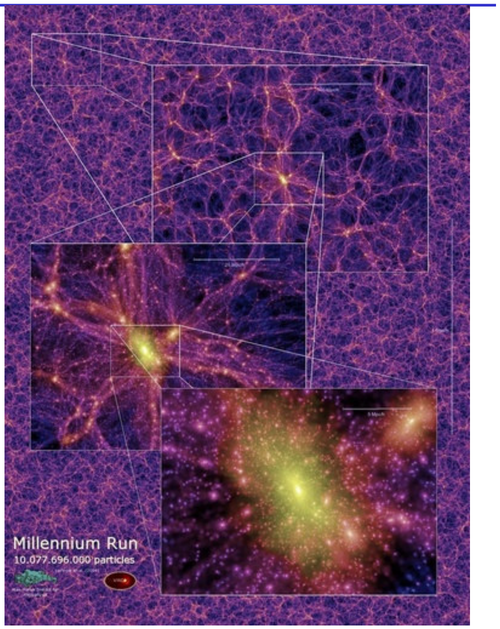
The Gadget code has been written and is maintained by the research group of Volker Springel at the Max Planck Institute for Astrophysics in Garching (Germany). It is a open source, freely available code for cosmological N-body/SPH simulations on massively parallel computers with distributed memory.

Gadget has been recently (2005) used to perform the Millenium Run, the largest N-Body cosmological simulation ever run.

Numerical simulations data. Examples :

1. The GADGET code data

#	Block ID	HDF5 identifier	Block content
1	HEAD		Header
2	POS	Coordinates	Positions
3	VEL	Velocities	Velocities
4	ID	ParticleIDs	Particle ID's
5	MASS	Masses	Masses (only for particle types with variable masses)
6	U	InternalEnergy	Internal energy per unit mass (only SPH particles)
7	RHO	Density	Density (only SPH particles)
8	HSML	SmoothingLength	SPH smoothing length h (only SPH particles)
9	POT	Potential	Gravitational potential of particles (only when enabled in makefile)
10	ACCE	Acceleration	Acceleration of particles (only when enabled in makefile)
11	ENDT	RateOfChangeOf Entropy	Rate of change of entropic function of SPH particles (only when enabled in makefile)
12	TSTP	TimeStep	Timestep of particles (only when enabled in makefile)



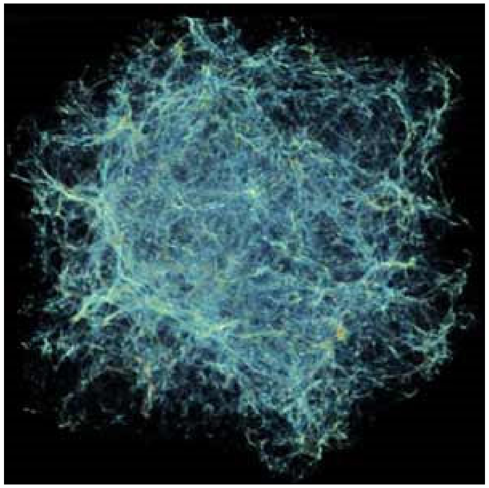
File Formats:

Gadget format

HDF5

Numerical simulations data. Examples:

2. The ENZO code and its data



File Format:

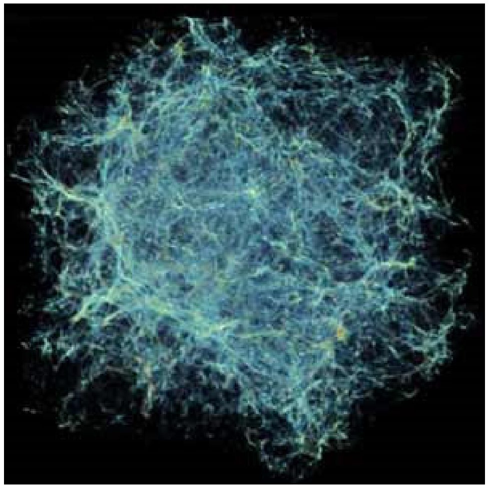
HDF5

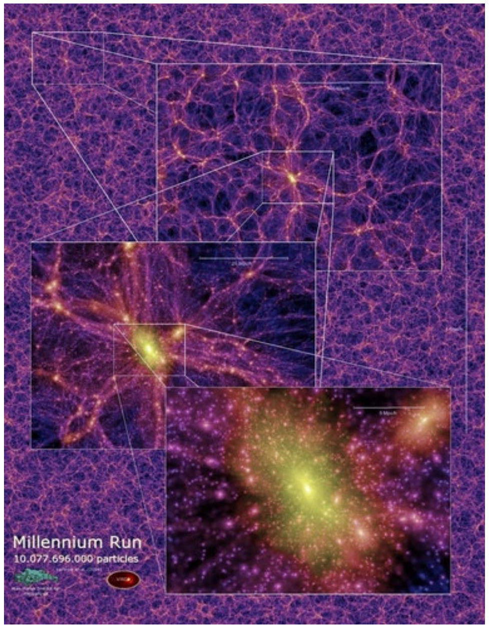
Enzo is an adaptive mesh refinement (AMR), grid-based hybrid code (hydro + N-Body) which is designed to do simulations of cosmological structure formation. Enzo was originally written by Greg Bryan under the supervision of Michael Norman while at the National Center for Supercomputing Applications at the University of Illinois. Enzo's home is the Laboratory for Computational Astrophysics at the Center for Astrophysics and Space Sciences, located at the University of California in San Diego. The code can be downloaded from the web site (http://cosmos.ucsd.edu/enzo).

DATA

Dark_Matter_Density	Dataset {400, 400, 400}
Density	Dataset {400, 400, 400}
Gas_Energy	Dataset {400, 400, 400}
Temperature	Dataset {400, 400, 400}
Total_Energy	Dataset {400, 400, 400}
particle_index	Dataset {64000000}
particle_mass	Dataset {64000000}
particle_position_x	Dataset {64000000}
particle_position_y	Dataset {64000000}
particle_position_z	Dataset {64000000}
particle_velocity_x	Dataset {64000000}
particle_velocity_y	Dataset {64000000}
particle_velocity_z	Dataset {64000000}
x-velocity	Dataset {400, 400, 400}
y-velocity	Dataset {400, 400, 400}
z-velocity	Dataset {400, 400, 400}

Cosmological simulations metadata





Metadata proposed

The description of the simulation data requires the following data:

Data describing the physical model:

Physical modules (Hydro, Dark Matter, Gravitation)
Cooling
Heating and Preheating
Star formation

Data describing the algorithm:

Spatial order of accuracy
Type of spatial interpolation
Temporal order of accuracy
Temporal integrator
Time stepping
Courant number

Data describing the grid:

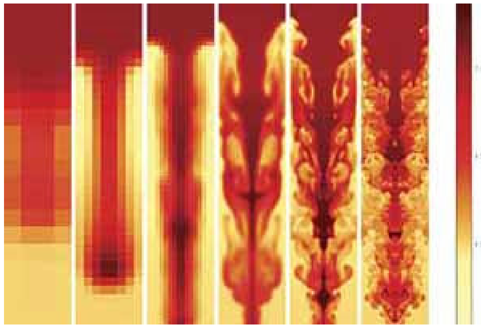
Grid vs. Particles
Number of particles or nuber of grid points in each coordinate direction
Physical coordinates of grid points (for each direction)
Type of boundary conditions

Physical parameters

Hubble constant
Density parameters
Initial fluctuation spectrum

Numerical simulations data. Examples:

3. MHD Simulations



PLUTO developed in collaboration between Osservatorio Astronomico di Torino and University of Torino (Mignone, Massaglia & Bodo 2004)

FLASH developed at the University of Chicago (Fryxell et al. 2000).

Both are AMR codes for simulations of general astrophysical flows that includes many physical modules (classical compressible hydrodynamics and magnetohydrodynamics (MHD) and relativistic hydrodynamics and MHD) and can make use of different coordinates (cartesian, cylindrical, spherical).

Metadata proposed

The description of the simulation data requires the following data:

Data describing the physical model:

Physical module (Hydro, MHD, Relativistic hydro, Relativistic MHD)

Inclusion of radiative cooling

Type of cooling

Data describing the algorithm:

Spatial order of accuracy

Type of spatial interpolation

Temporal order of accuracy

Temporal integrator

Riemann solver

Courant number

Magnetic monopole control method (for MHD)

Data describing the grid:

Coordinate type (cartesian, cylindrical spherical ...)

Number of grid points in each coordinate direction

Physical coordinates of grid points (for each direction)

Type of boundary conditions

Physical parameters

Mach number

Density ratio between jet and ext. medium

Jet velocity (relativistic)

Plasma b (MHD)}

z-velocity

Numerical simulations data.

Data model 1.

A Simulation Data Model (hereafter DM) is a way of describing an astrophysical simulation. The DM is a way to provide a conceptual, logical and **interoperable** description of a simulation. It is not tied to the way data providers internally store, describe, manage or organize their archives.

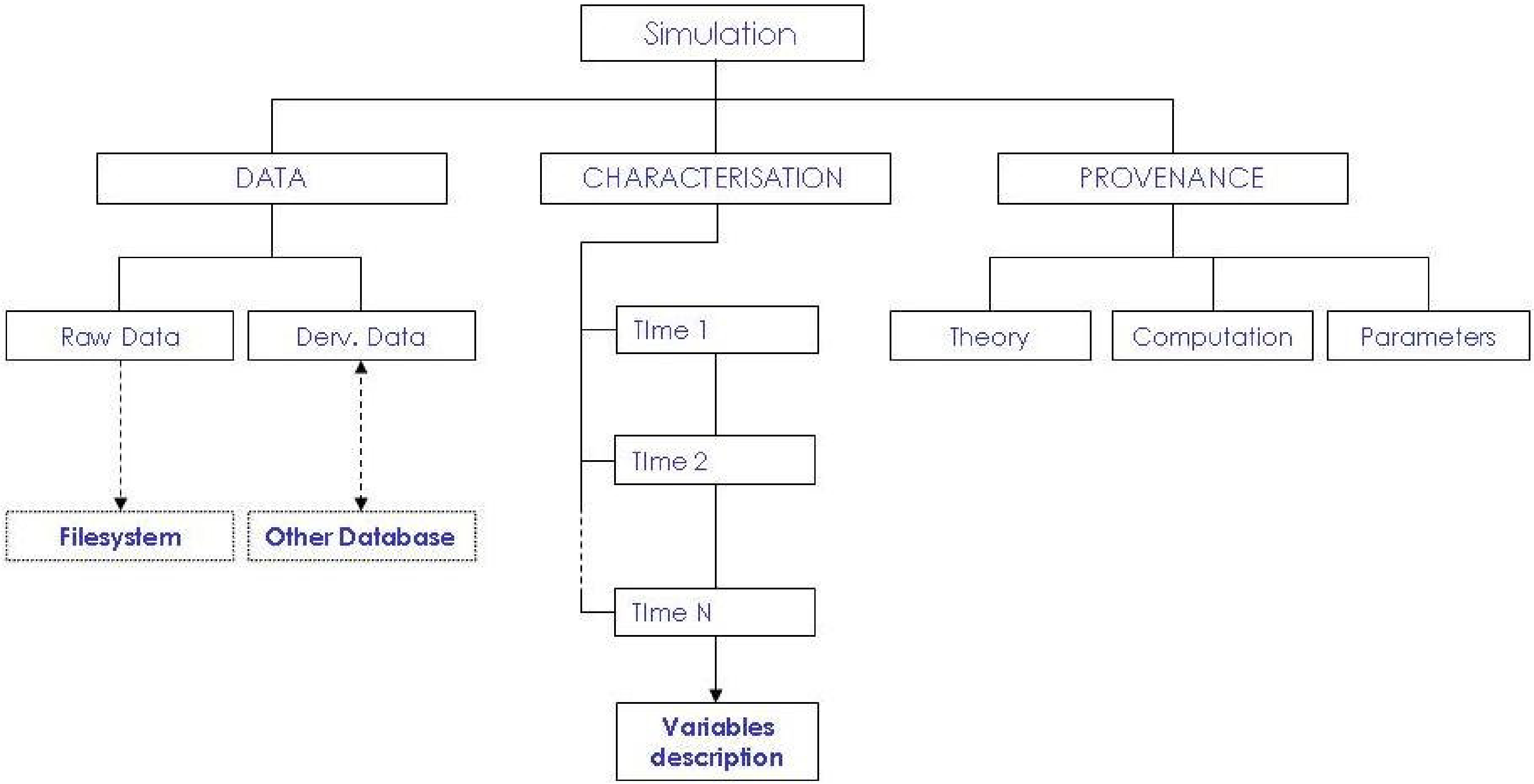
We propose a data model for simulation data providing a general architecture which encompasses all kind of simulations.

The architecture is hierarchical

The root node of the hierarchy (Level 0), our basic object, is the **Simulation** (Level 0 of hierarchy). DM must characterize completely the Simulation.

Numerical simulations data.

Data Model

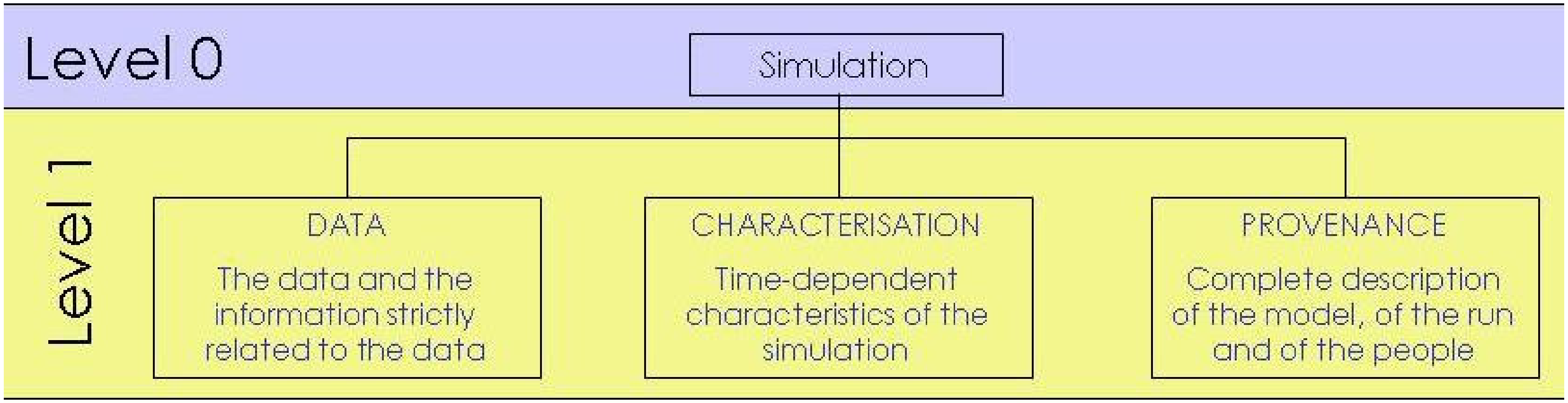


Cambridge, February 2006

Numerical simulations data.

Data model 2.

Starting from the Observation model and according to pub1, also for Simulations we can define three main classes: **Data, Characterization and Provenance.** These three classes, which compose the '**Level 1**' of the DM, are further specified in subclasses in a hierarchical pattern.



Level 0

Simulation

Level 1

DATA

The data and the information strictly related to the data

CHARACTERISATION

Time-dependent characteristics of the simulation

PROVENANCE

Complete description of the model, of the run and of the people

Numerical simulations data.

Data

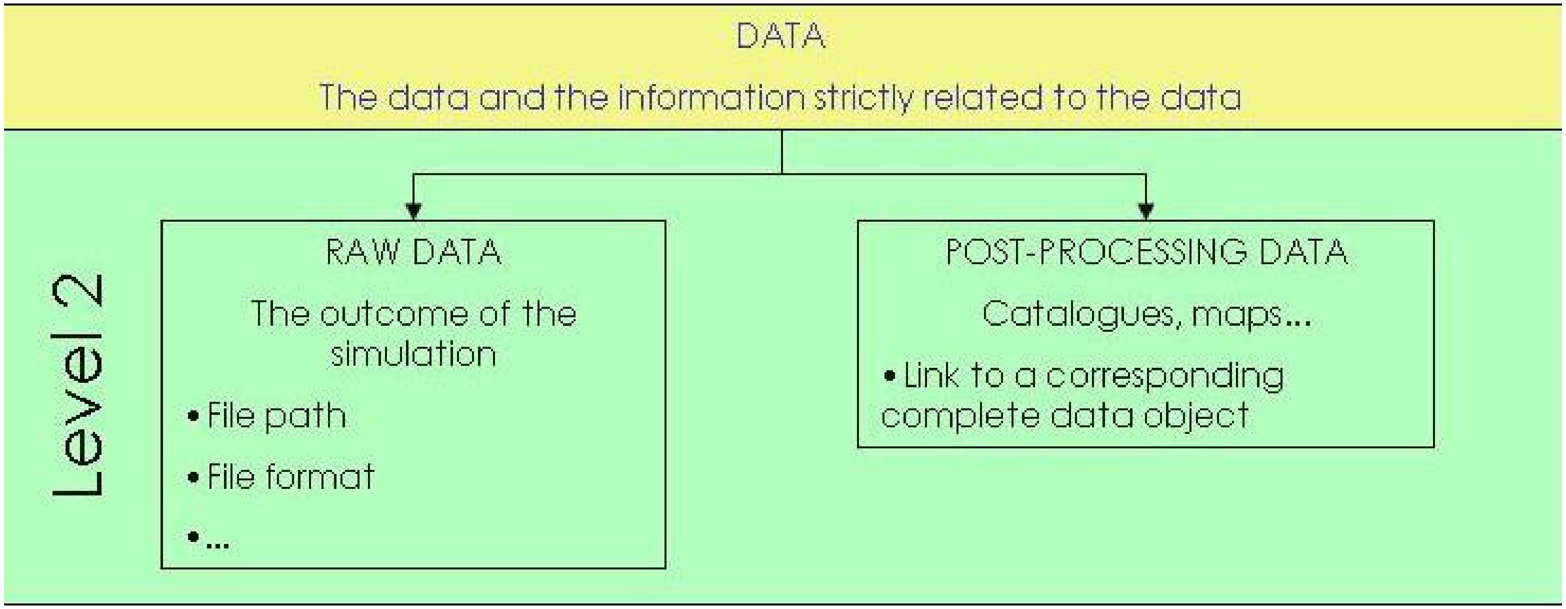
Raw data are annotated by a number of Level 2 metadata. The purpose of these metadata is to keep track of a file location and its format characteristics. In particular:

•Link to the physical location of the file in the filesystem

•File format

•…

Details about its content are managed by the Characterization class.



DATA

The data and the information strictly related to the data

Level 2

RAW DATA

The outcome of the simulation

- File path
- File format
- …

POST-PROCESSING DATA

Catalogues, maps…

- Link to a corresponding complete data object

Numerical simulations data.

Characterization

This is the item that **most differs** from the corresponding observational one.

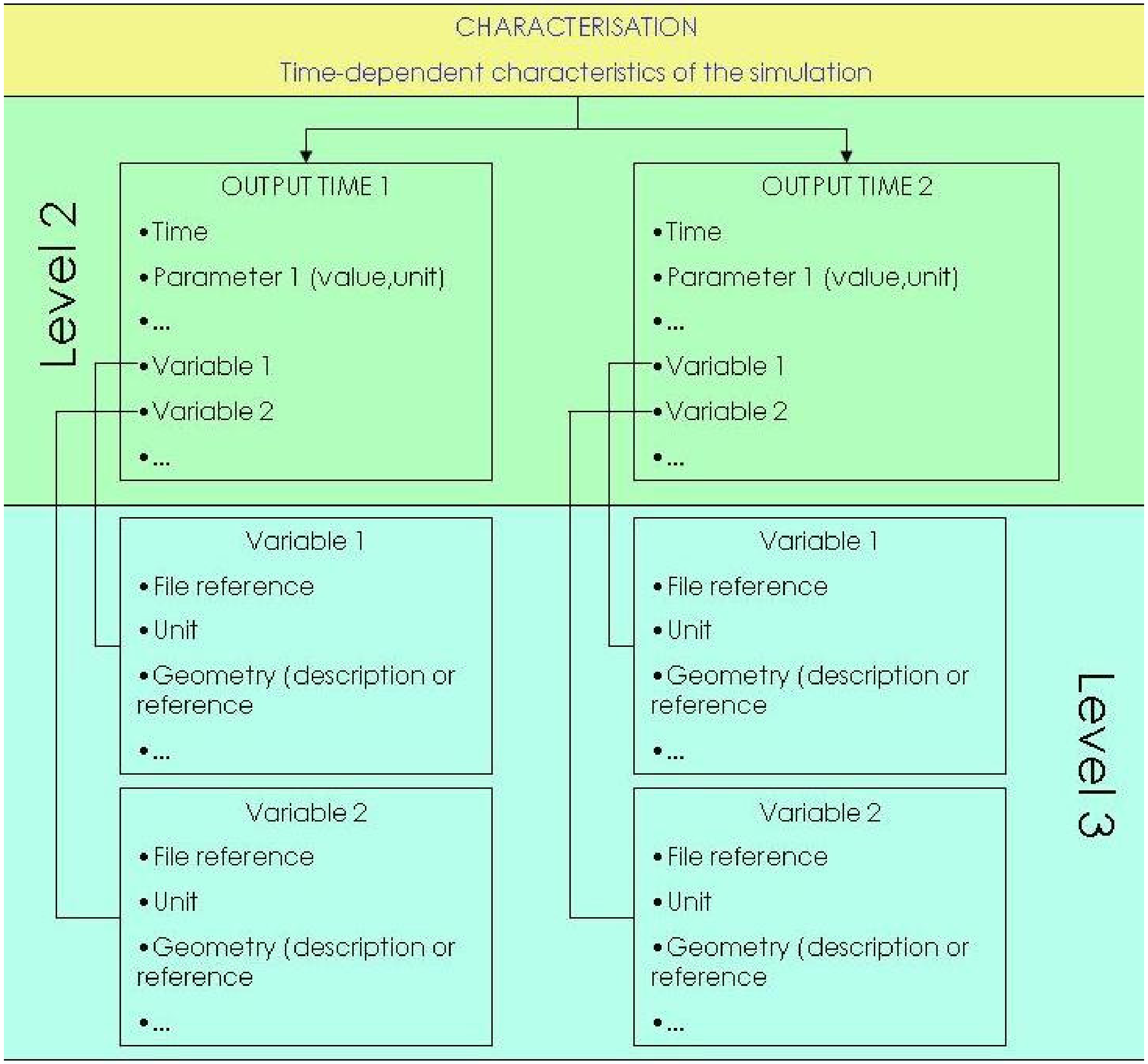
It handles the concept of time.
It fully describes variables at each output time

Characterization parameters (in general, a value-unit pair) can be either global (the output time or some kind of control data - averages, statistics…) or detailed information about each output variable (which characterize completely the variable itself) like:

- The reference to the corresponding entry in the Data class (a variable BELONGS to a file)
- The variable type (cubic grid, AMR grid, particles)
- The variable size (number of elements)
- The description of the geometry and topology
- The unit
- The maximum and minimum spatial and mass (or other) resolutions
- …

Numerical simulations data.

Characterization (2)



CHARACTERISATION

Time-dependent characteristics of the simulation

Level 2

OUTPUT TIME 1

- Time
- Parameter 1 (value,unit)
- ...
- Variable 1
- Variable 2
- ...

OUTPUT TIME 2

- Time
- Parameter 1 (value,unit)
- ...
- Variable 1
- Variable 2
- ...

Level 3

Variable 1

- File reference
- Unit
- Geometry (description or reference
- ...

Variable 2

- File reference
- Unit
- Geometry (description or reference
- ...

Variable 1

- File reference
- Unit
- Geometry (description or reference
- ...

Variable 2

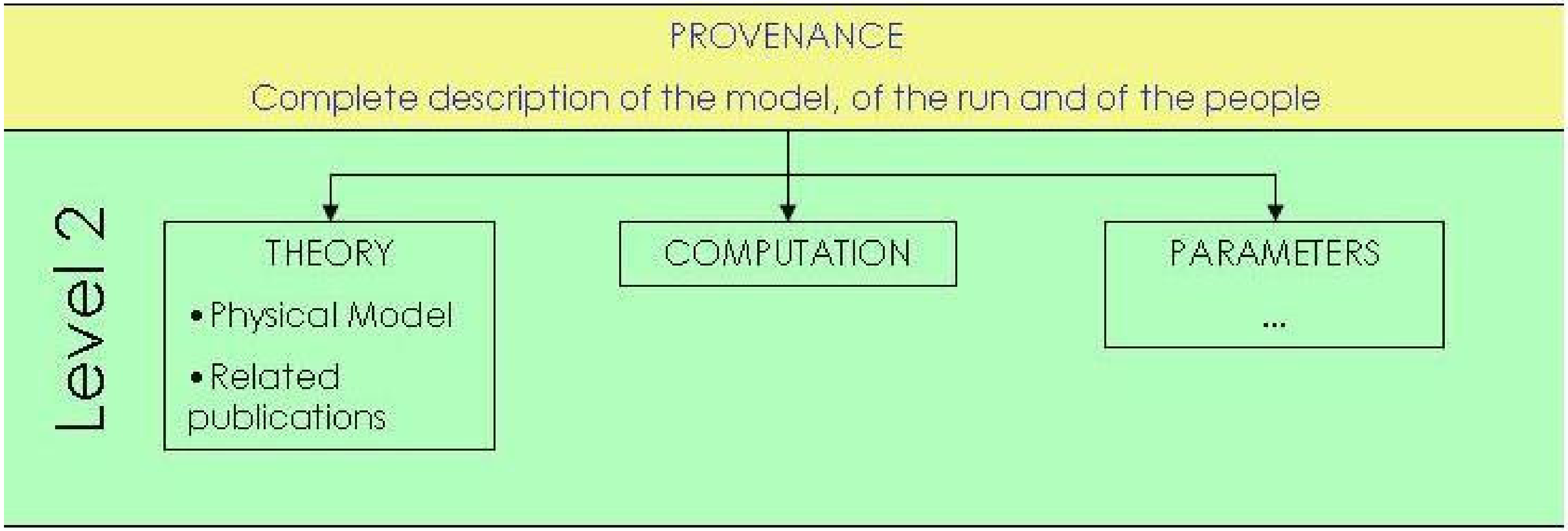
- File reference
- Unit
- Geometry (description or reference
- ...

Numerical simulations data.

Provenance

The Provenance object contains the information describing the simulation as a whole. The Provenance object is defined as 'the description of how the dataset was created' which for a simulation we are able to describe entirely. Two simulations with the same Provenance parameters are identical.

Provenance can be broken down into the Theory, Computation and Parameters.



PROVENANCE

Complete description of the model, of the run and of the people

Level 2

THEORY

- Physical Model
- Related publications

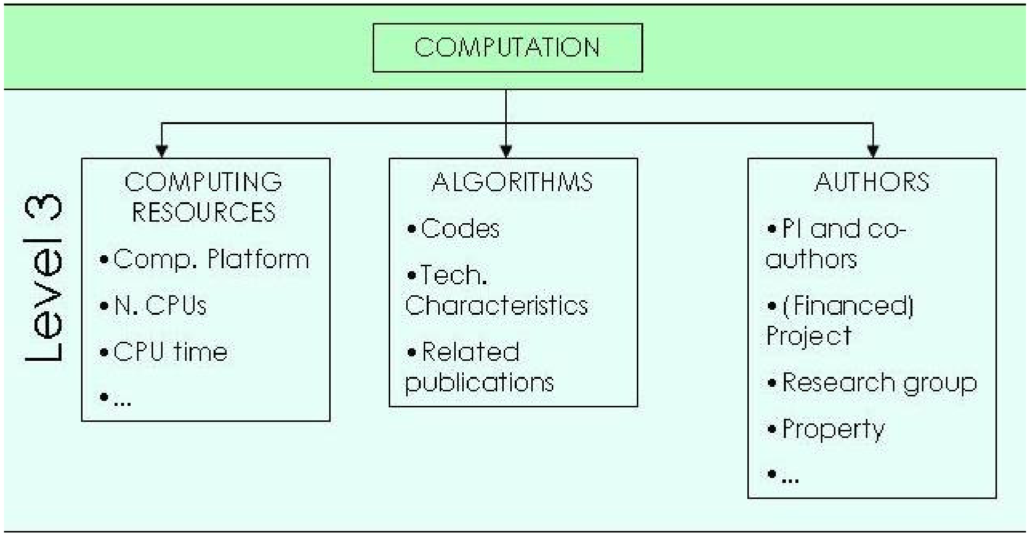
COMPUTATION

PARAMETERS

...

Numerical simulations data.

Provenance (2)



Level 3

COMPUTATION

COMPUTING RESOURCES

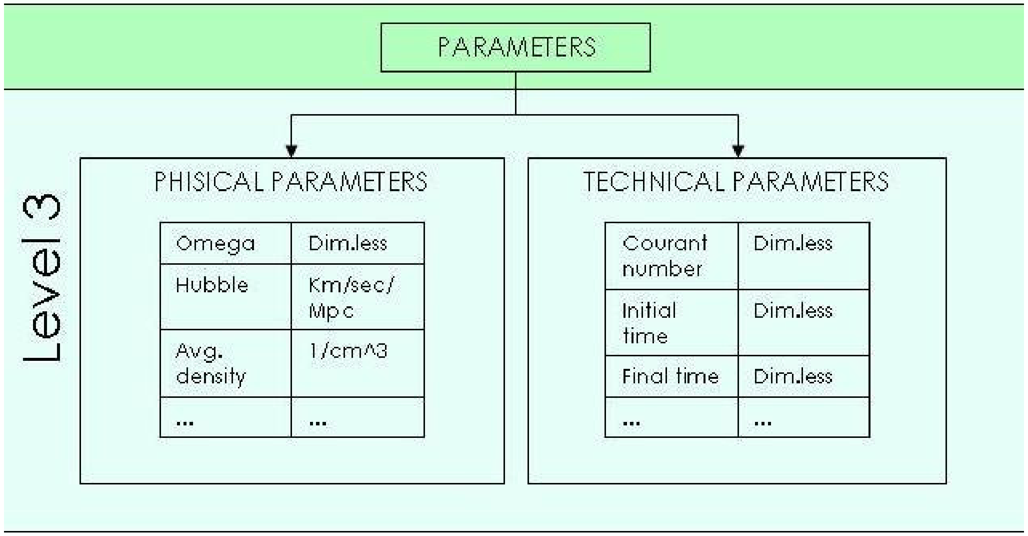
- Comp. Platform
- N. CPUs
- CPU time
- ...

ALGORITHMS

- Codes
- Tech. Characteristics
- Related publications

AUTHORS

- PI and co-authors
- (Financed) Project
- Research group
- Property
- ...



Level 3

PARAMETERS

PHISICAL PARAMETERS

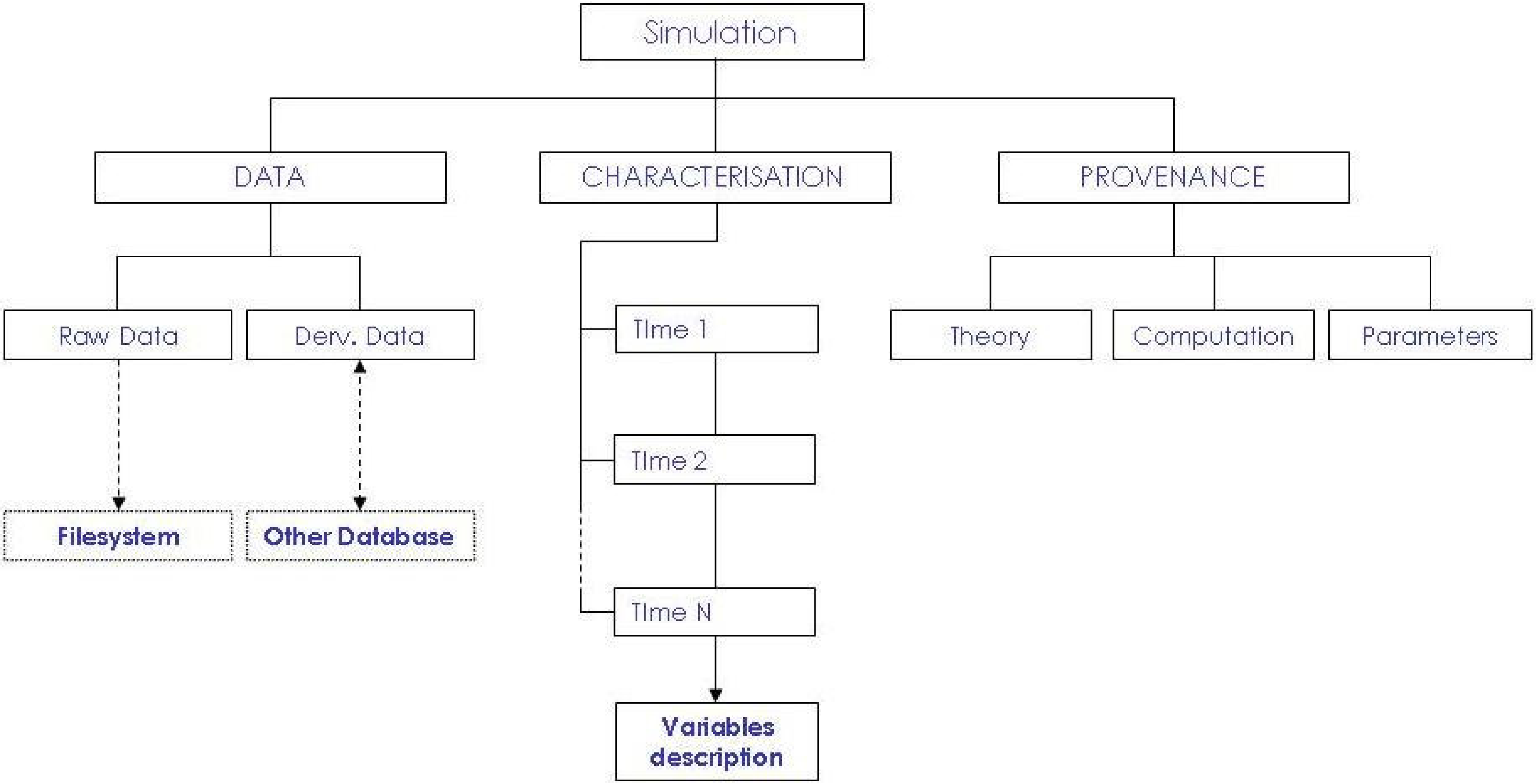
Omega	Dim.less
Hubble	Km/sec/ Mpc
Avg. density	1/cm^3
...	...

TECHNICAL PARAMETERS

Courant number	Dim.less
Initial time	Dim.less
Final time	Dim.less
...	...

Numerical simulations data.

Summary


Simulation
DATA
CHARACTERISATION
PROVENANCE
Raw Data
Derv. Data
Filesystem
Other Database
TIme 1
TIme 2
TIme N
Variables description
Theory
Computation
Parameters