UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

GROMACS, XML and Python

David van der Spoel

Uppsala University

FSatom
Lyon
8-10-2003

UPPSALA UNIVERSITET

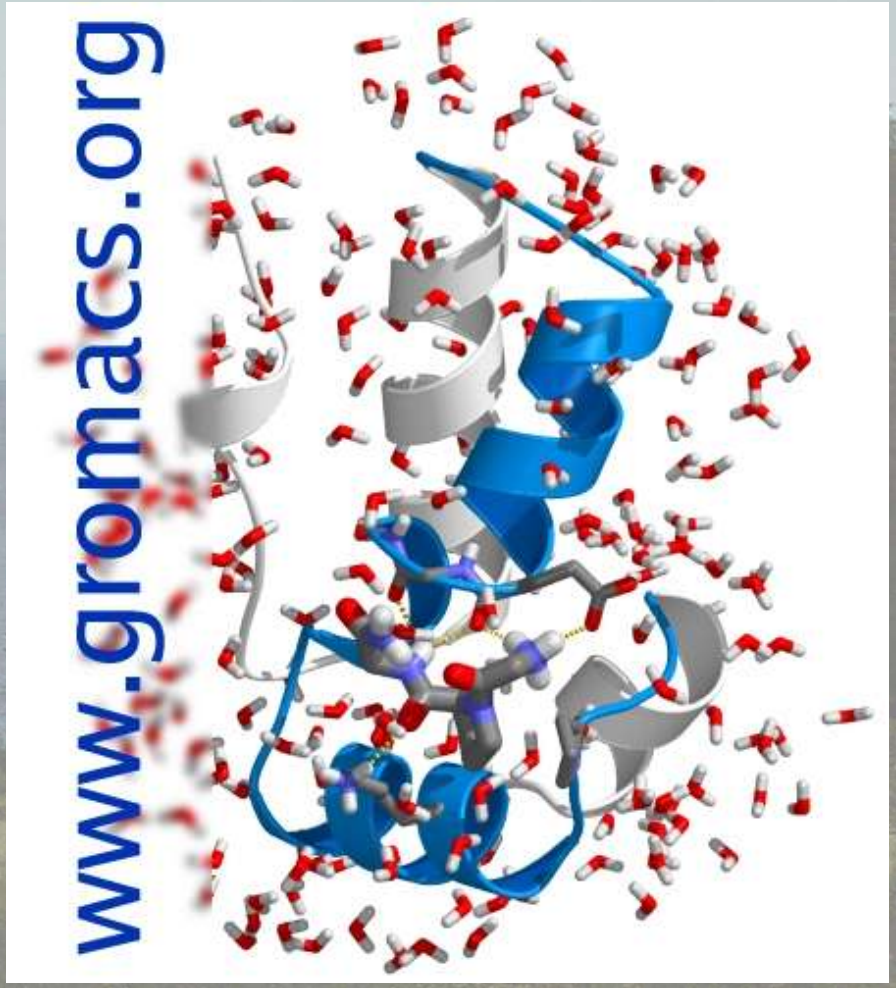
David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

GROMACS

- An efficient and versatile package for molecular simulation.
- GPL License
- 1000+ users worldwide
- J. Mol. Mod 7 (2001) pp. 306-317



UPPSALA
UNIVERSITET

David van der Spoel

Molecular
Biophysics Group,
Department of Cell
and Molecular
Biology

spoel at
xray.bmc.uu.se

Molecular simulation

- Given the atomic coordinates of a set of molecules, compute energy and forces according to a classical Hamiltonian.
- Integrate Newton's equations of motion, with a timestep of 1-2 fs.
- Repeat for 10^6 - 10^9 steps.
- Analyse the results.

UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Example simulation

- 50 octanol + 3650 water molecules
- Periodic box of 5^3 nm
- Particle mesh Ewald algorithm for Coulomb interactions
- 10 ns simulation
- CPU time 9 days on dual P3/866

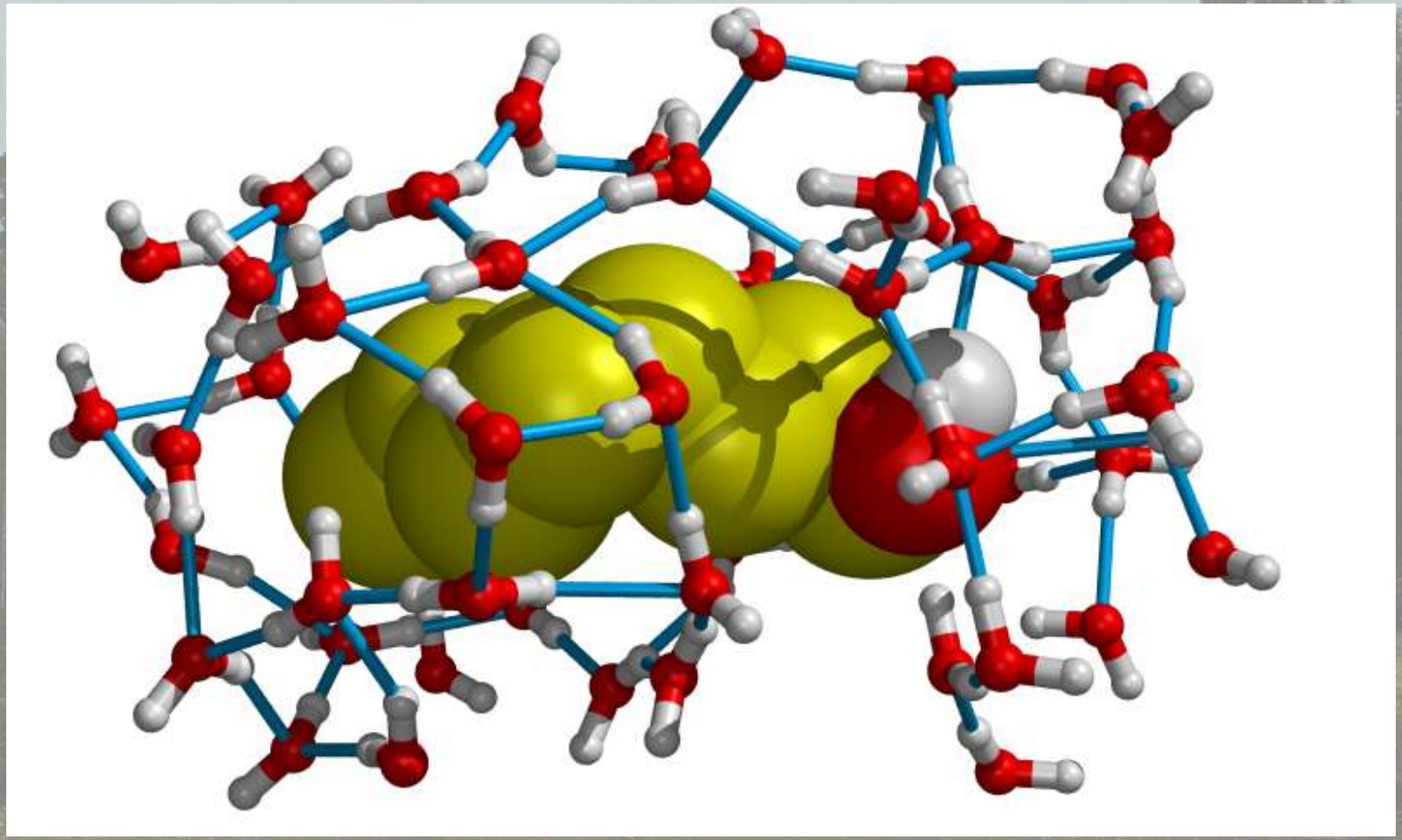
UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Octanol solvation



FSatom
Lyon
8-10-2003

Octanol solvation

UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

FSatom
Lyon
8-10-2003

UPPSALA
UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

GPL License

- Derived software must be GPL too
- No one can steal the source and make money on it
- Easy to use and incorporate other GPL software, e.g. FFTW (http://www.fftw.org) and LAM (http://www.lam-mpi.org)
- Public money should benefit the taxpayer

UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

1000+ users worldwide

- A few thousand downloads, roughly 1000 people on mailing lists
- Users on all continents
- New software for the folding at home project (Protein Folding, see http://folding.stanford.edu)

UPPSALA
UNIVERSITET

David van der Spoel

Molecular
Biophysics Group,
Department of Cell
and Molecular
Biology

spoel at
xray.bmc.uu.se

Software design

- C: 230,000 lines of C source code (of which 42,000 lines *generated*)
- S: 130,000 lines of Assembly code
- F: 52,000 lines of *generated* Fortran code
- Quasi object oriented (strong typing etc.)
- User friendly command line interface
- 200+ pages user manual

UPPSALA
UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Software tools

- CVS
- Autoconf/automake
- PHP (web)
- XML
- Python

UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

GROMACS Library

- Many functions for file I/O, including automatic determining of floating point precision
- Platform-independent binary files (XDR)
- Functions for handling user input, e.g.:

```
parse_common_args(argc,argv,...);
```

- Built-in documentation: the user documentation is part of each program and can be printed with a `-h` option (in HTML, LaTex or nroff format)

UPPSALA
UNIVERSITET

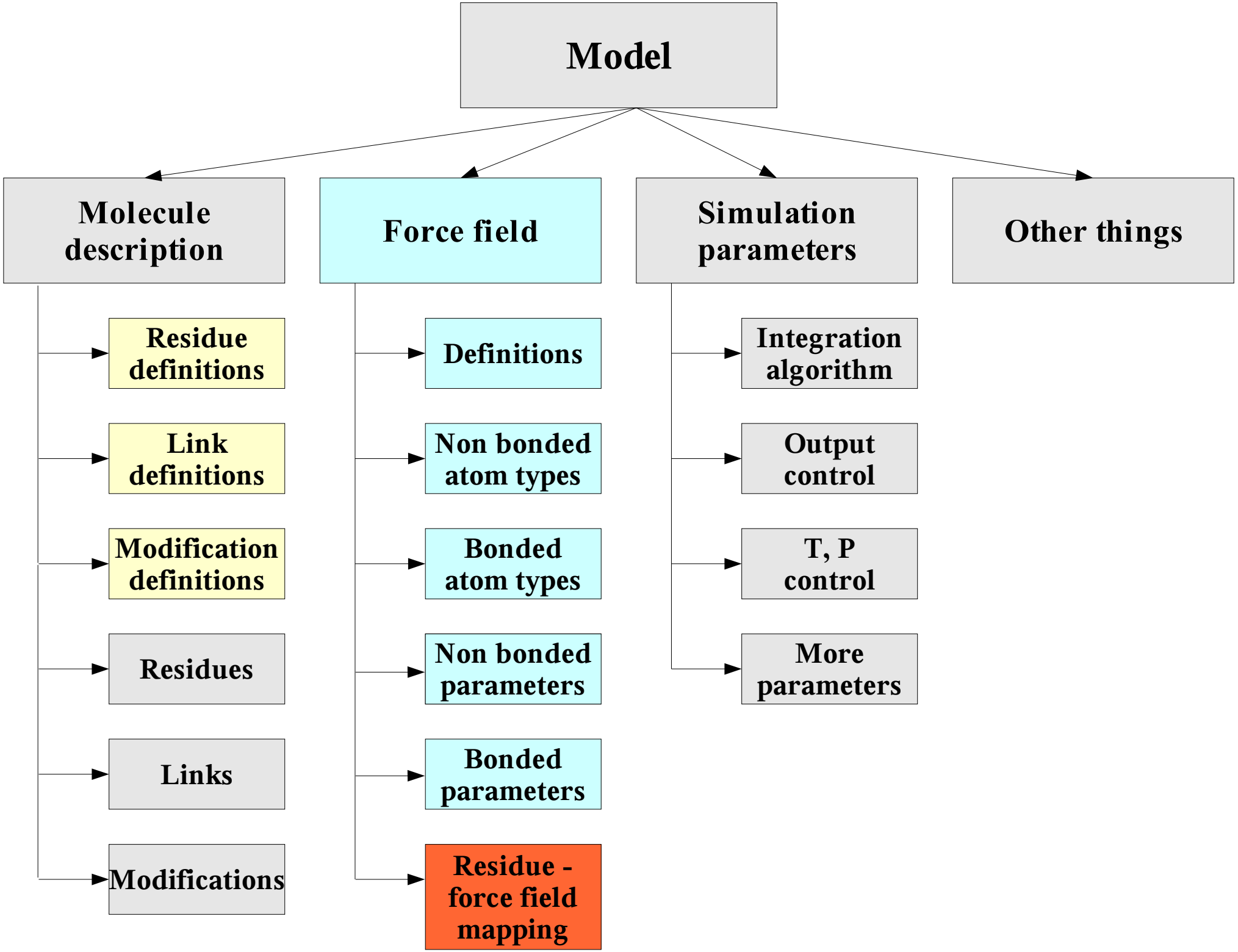
David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Why use XML?

- Growing complexity of input files
- Self contained file format which can/could be validated
- Separate data from algorithms (anti object oriented!)
- Extensible?
- Machine readable (using e.g. libxml2)


Model
Molecule description
Force field
Simulation parameters
Other things
Residue definitions
Link definitions
Modification definitions
Residues
Links
Modifications
Definitions
Non bonded atom types
Bonded atom types
Non bonded parameters
Bonded parameters
Residue - force field mapping
Integration algorithm
Output control
T, P control
More parameters

```
<!ELEMENT macromolecules (residues,macromolecule*)>

<!ELEMENT residues (residue*,linkdef*,moddef*)>

<!ELEMENT residue (ratom+,rbond*,rangle*,
                   rimproper*,rdihedral*,raddh*)>
<!ATTLIST residue
          restype  ID    #REQUIRED
          longname CDATA #REQUIRED>

<!-- We can not use ID here for atom names since  -->
<!-- there is no hope to make them unique, e.g.   -->
<!-- all amino acids have the same name for       -->
<!-- backbone atoms.                              -->
<!ELEMENT ratom EMPTY>
<!ATTLIST ratom
          name  CDATA #REQUIRED>
```

```
<!-- Bond angle to define angle vibrations   -->
<!ELEMENT rangle EMPTY>
<!ATTLIST rangle
          a1 CDATA #REQUIRED
          a2 CDATA #REQUIRED
          a3 CDATA #REQUIRED>

<!-- Out of plane dihedral angle             -->
<!ELEMENT rimproper EMPTY>
<!ATTLIST rimproper
          a1 CDATA #REQUIRED
          a2 CDATA #REQUIRED
          a3 CDATA #REQUIRED
          a4 CDATA #REQUIRED>

<!-- Normal torsion angle                    -->
<!ELEMENT rdihedral EMPTY>
<!ATTLIST rdihedral
          a1 CDATA #REQUIRED
          a2 CDATA #REQUIRED
          a3 CDATA #REQUIRED
          a4 CDATA #REQUIRED>
```

```
<!-- Definition of how to add (hydrogen) atoms to,  -->
<!-- molecules using a certain geometry             -->
<!-- (see GROMACS manual)                           -->
<!ELEMENT raddh EMPTY>
<!ATTLIST raddh
          hclass   ( polar | aromatic | aliphatic )
          addgeom  CDATA #REQUIRED
          addnum   CDATA #REQUIRED
          addto    CDATA #REQUIRED>
```

```
<!-- Define a covalent link between two residues, -->
<!-- given in restype and the atoms are given as  -->
<!-- well (atomprev and atomnext)                 -->
<!-- A reference distance between the atoms does  -->
<!-- also have to be given in nm (but that does   -->
<!-- not have to be used).                        -->
<!ELEMENT linkdef EMPTY>
<!ATTLIST linkdef
          linktype ID       #REQUIRED
          restype  IDREFS   #REQUIRED
          atomprev CDATA    #REQUIRED
          atomnext CDATA    #REQUIRED
          refdist  CDATA    #REQUIRED>
```

```
<!-- Description of termini, etc. or maybe other   -->
<!-- modifications too. It is important that the   -->
<!-- implementation deletes first, and adds atoms  -->
<!-- afterwards                                    -->
<!ELEMENT moddef (moddelete*,modadd*,modrepl*)>
<!ATTLIST moddef
          modtype ID    #REQUIRED>
<!ELEMENT moddelete EMPTY>
<!ATTLIST moddelete
          delname CDATA #REQUIRED>
<!ELEMENT modreplace EMPTY>
<!ATTLIST modreplace
          oldname CDATA #REQUIRED
          newname CDATA #REQUIRED>
<!ELEMENT modadd EMPTY>
<!ATTLIST modadd
          addname CDATA #REQUIRED
          addgeom CDATA #REQUIRED
          addto   CDATA #REQUIRED>
```

```
<!-- Generic stuff to define proteins etc. as  -->
<!-- built up from blocks such as amino acids  -->
<!-- and HEME etc.                             -->
<!ELEMENT macromolecule (mblock+,mlink*,mmod*)>
<!ATTLIST macromolecule
          mname ID #REQUIRED
          protonated ( no | polar | all ) "polar">
<!ELEMENT mblock EMPTY>
<!ATTLIST mblock
          resname   ID    #REQUIRED
          restype   IDREF #REQUIRED>
<!ELEMENT mlink EMPTY>
<!ATTLIST mlink
          resname    IDREFS #REQUIRED
          mlinktype  IDREF #REQUIRED>
<!ELEMENT mmod EMPTY>
<!ATTLIST mmod
          resname IDREF #REQUIRED
          modtype IDREF #REQUIRED>

<!-- End of DTD                                -->
```

```
<?xml version="1.0" encoding="ISO-8859-1"
       standalone="yes"?>

<!DOCTYPE macromolecules PUBLIC "residues.dtd"
          "residues.dtd">

<macromolecules>
<residues>
  <residue restype="AAA" longname="any_residue">
    <ratom name="X"/>
  </residue>

  <residue restype="ALA">
    <ratom name="N">
    <ratom name="H">
    <ratom name="CA">
    <ratom name="HA">
    <ratom name="CB">
    <ratom name="HB1">
    <ratom name="HB2">
    <ratom name="HB3">
    <ratom name="C">
    <ratom name="O">
```

```
<rbond a1="N" a2="H"/>
<rbond a1="N" a2="CA"/>
<rbond a1="CA" a2="HA"/>
<rbond a1="CA" a2="CB"/>
<rbond a1="CA" a2="C"/>
<rbond a1="CB" a2="HB1"/>
<rbond a1="CB" a2="HB2"/>
<rbond a1="CB" a2="HB3"/>
<rbond a1="C" a2="O"/>
<rbond a1="-C" a2="N"/>
```

```
  <rimproper a1="N" a2="-C" a3="CA" a4="H"/>
  <rimproper a1="-C" a2="-CA" a3="N" a4="-O"/>
  <rimproper a1="CA" a2="N" a3="C" a4="CB"/>

  <raddh hclass="polar"     addgeom="1" addnum="1"
         addto="N -C CA"/>
  <raddh hclass="aliphatic" addgeom="5" addnum="1"
         addto="CA N C CB"/>
  <raddh hclass="aliphatic" addgeom="4" addnum="3"
         addto="CB CA N"/>
</residue>

<!-- More residues to be added here of course! -->
```

```
<linkdef linktype="peptide" restype="AAA AAA"
         atomprev="C" atomnext="N" refdist="0.133"/>
<linkdef linktype="disulphide" restype="CYS CYS"
         atomprev="SG" atomnext="SG" refdist="0.2"/>
```

```
  <moddef modtype="NH3+"
    <modreplace oldname="N" newname="N"/>
    <modreplace oldname="CA" newname="CA"/>
    <modadd addname="H" addgeom="4" addto="N CA C"/>
    <moddelete delname="H"/>
  </moddef>

  <moddef modtype="COO-"
    <modreplace oldname="C" newname="C"/>
    <modadd addname="O" addgeom="3" addto="C CA N"/>
    <moddelete delname="O"/>
  </moddef>

</residues>
```

```
<macromolecule mname="AP" protonated="all">
  <mblock resname="A1" restype="ALA" />
  <mblock resname="P2" restype="PRO" />
  <mlink resname="A1 P2" mlinktype="peptide" />
  <mmod resname="A1" modtype="NH3" />
  <mmod resname="P2" modtype="COO" />
</macromolecule>
</macromolecules>
```

UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Is this the right approach?

- DTD versus Schema
- Storing information in attributes
- Transformations using XSLT?
- Use CML?

UPPSALA
UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Why use Python?

- Easy to program
- Portable
- Can be used for GUI design using several underlying libraries (wxWindows, FLTK, PyGTK, Qt-Python etc.)
- Can be used as a generic scripting language (rather than a dedicated scripting language)

UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Python in GROMACS...

- In the first hand for developing a GUI
- Using tools/libraries in a simple manner (NetCDF/XML)
- Interfacing with other programs (MMTK, pymol, VMD)

UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Python in GROMACS...

- Use the GROMACS library function `parse_common_args` to generate python scripts
- A simple generic dialog box routine using Tkinter

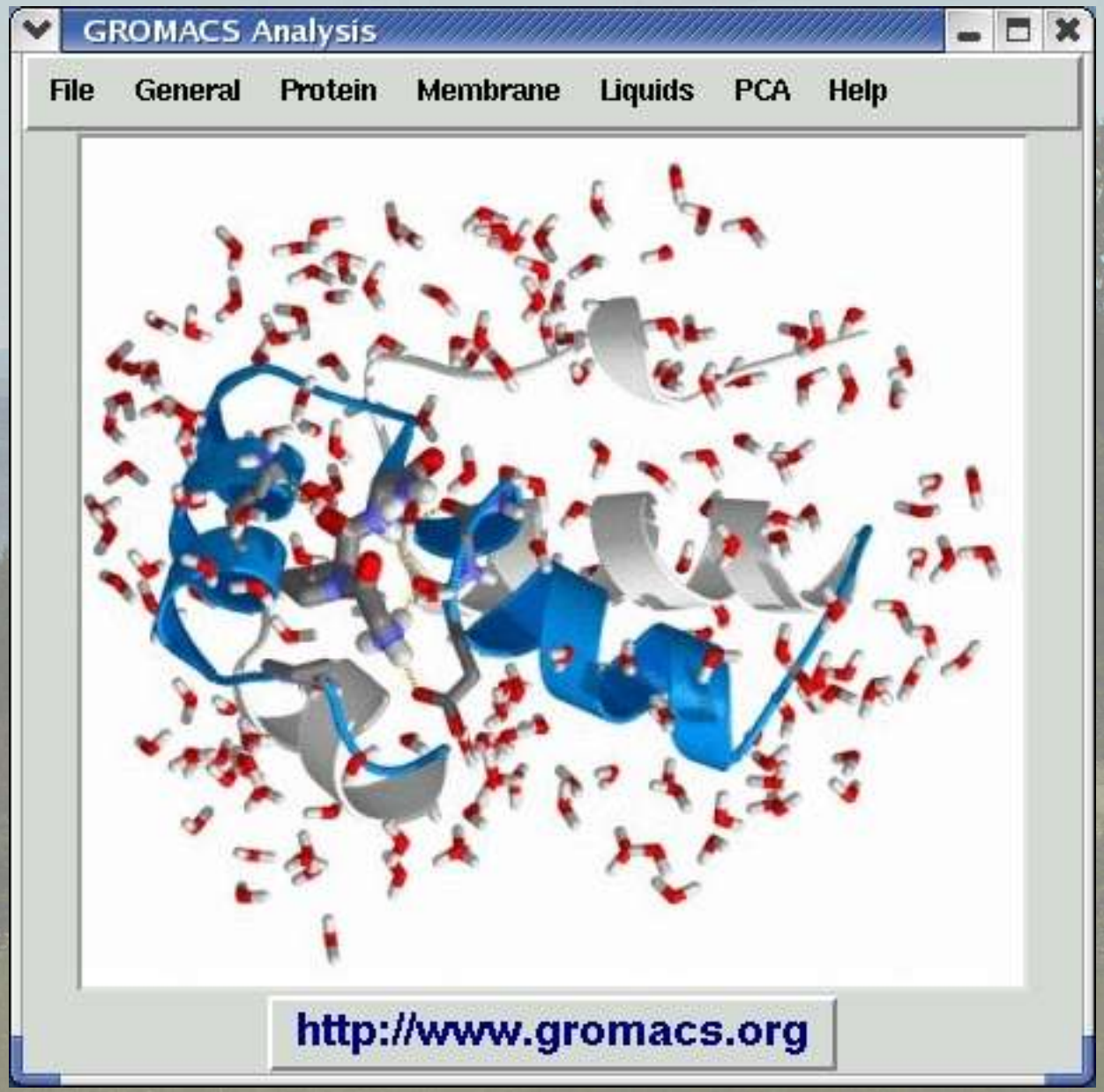
UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Analysis front-end



UPPSALA
UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Example application

- trjconv: convert and process molecular dynamics trajectories

UPPSALA
UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Summary

- Use someone else's programs, but if you must write your own, distribute it under an open source license
- The happiest programs are programs that write programs
- Use the right tool for the problem (Python is the answer, but what was the question?)
- XML Now! (but how?)

UPPSALA UNIVERSITET

David van der Spoel

Molecular Biophysics Group, Department of Cell and Molecular Biology

spoel at xray.bmc.uu.se

Acknowledgements

- Berk Hess (Groningen)
- Erik Lindahl (Stanford)
- Anton Feenstra (Amsterdam)
- Herman Berendsen (Groningen)
- Konrad Pywowarczik (Krakow)
- Michiel van Lun (Uppsala)