Plugins in ls1 mardyn

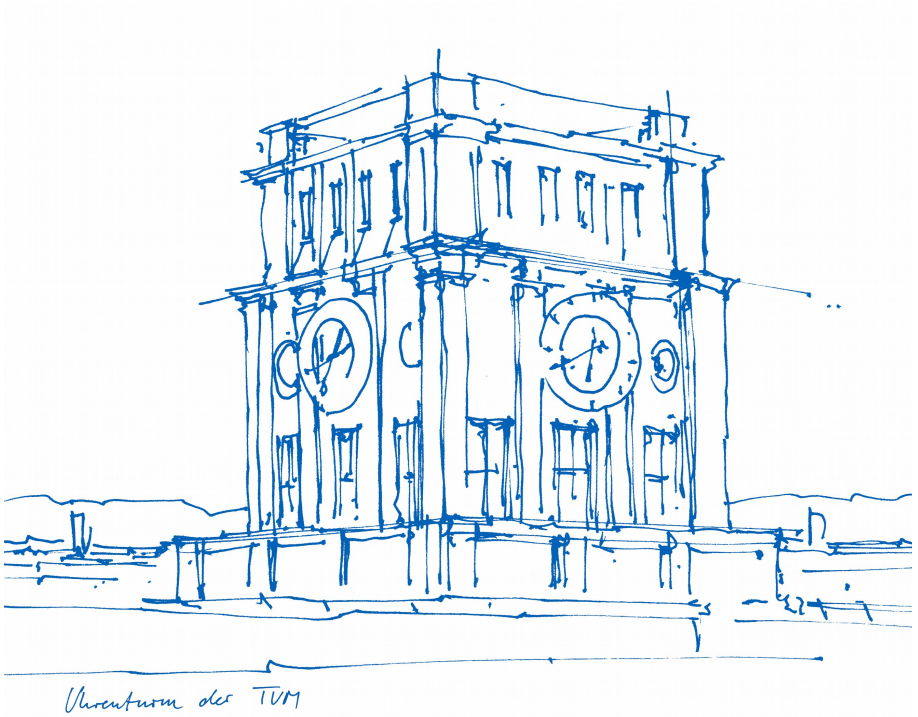
Steffen Seckler

Technical University of Munich

Faculty of Informatics

SCCS

Kaiserslautern, 12.10.2018

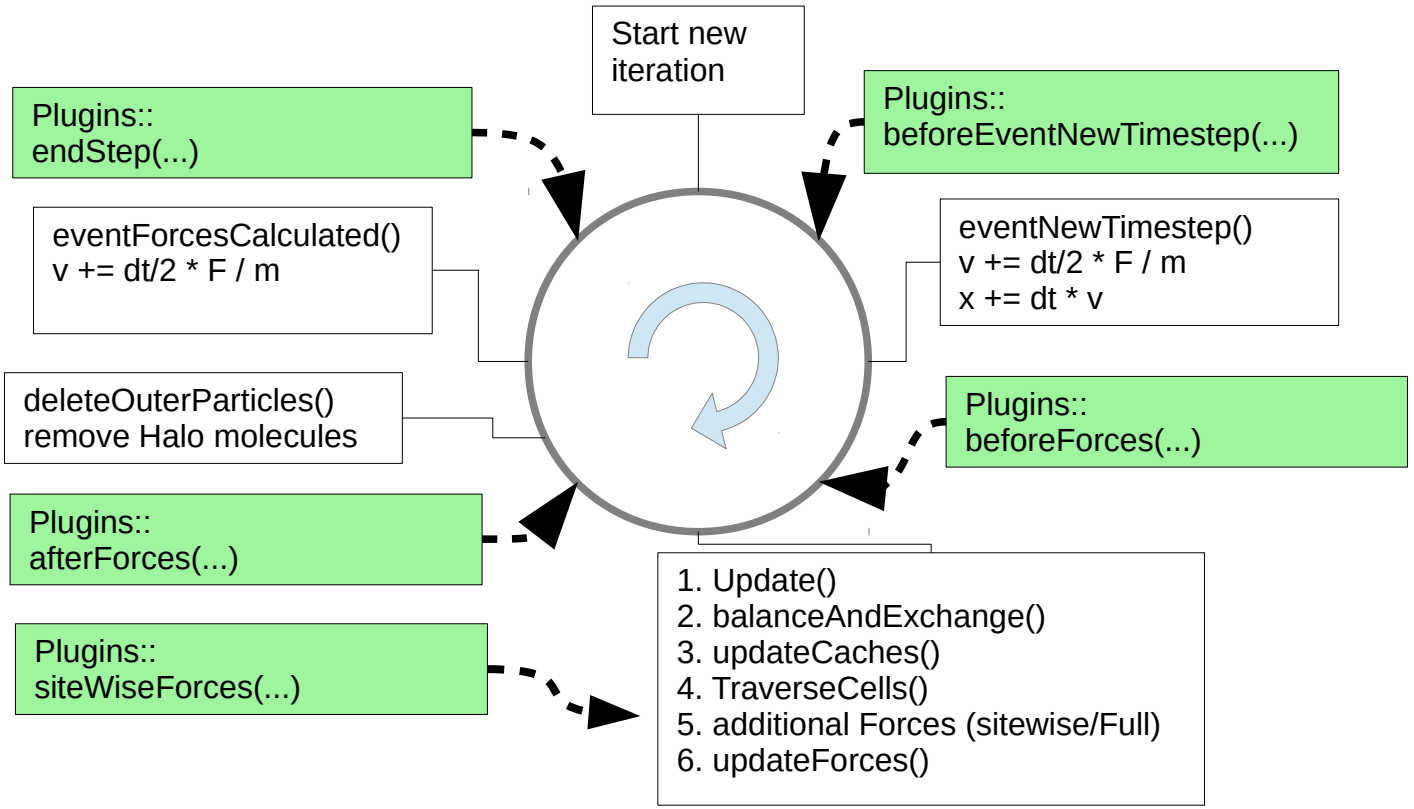


Plugins – Motivation

- **Make everything easy:**
  - Easy enabling / disabling
  - Easy reusability
  - Easy implementation
  - Easy debugging
  - Easier overview of features

- **Benefits**:
  - Less knowledge about code is required to write a plugin
  - Good isolation of code:
    - Normal code remains untouched
    - XML config is easy
    - Unit testing of plugins is easy

Overview


Start new iteration
Plugins:: endStep(...)
Plugins:: beforeEventNewTimestep(...)
eventForcesCalculated() v += dt/2 * F / m
eventNewTimestep() v += dt/2 * F / m x += dt * v
deleteOuterParticles() remove Halo molecules
Plugins:: beforeForces(...)
Plugins:: afterForces(...)
Plugins:: siteWiseForces(...)
1. Update()
2. balanceAndExchange()
3. updateCaches()
4. TraverseCells()
5. additional Forces (sitewise/Full)
6. updateForces()

Extension Points

Function / Extension Point	remarks	E.g.
beforeEventNew Timestep	Normally use endStep(…) instead.	MirrorSystem
beforeForces	Positions updated! → You can force positions here! (only small moves!) Molecules are not in the correct cells! (so don‘t use regionIterator)	COMaligner
siteWiseForces	Calculation of forces on molecule sites. Use this for Fljcenteradd() etc..	WallPotential
afterForces	Halo is present, use for Fadd() or if you need information of halo molecules.	HaloParticleWriter
endStep	Velocity is changed. Most plugins go here!	Output plugins

Steps to write a new Plugin

1) Check whether the plugin already exists!
2) Derive new class from PluginBase.
3) Write your code + document it using doxygen.
4) Add your code to the PluginFactory:

   #include „plugins/PLUGINNAME.h“

   REGISTER_PLUGIN(NewPlugin);

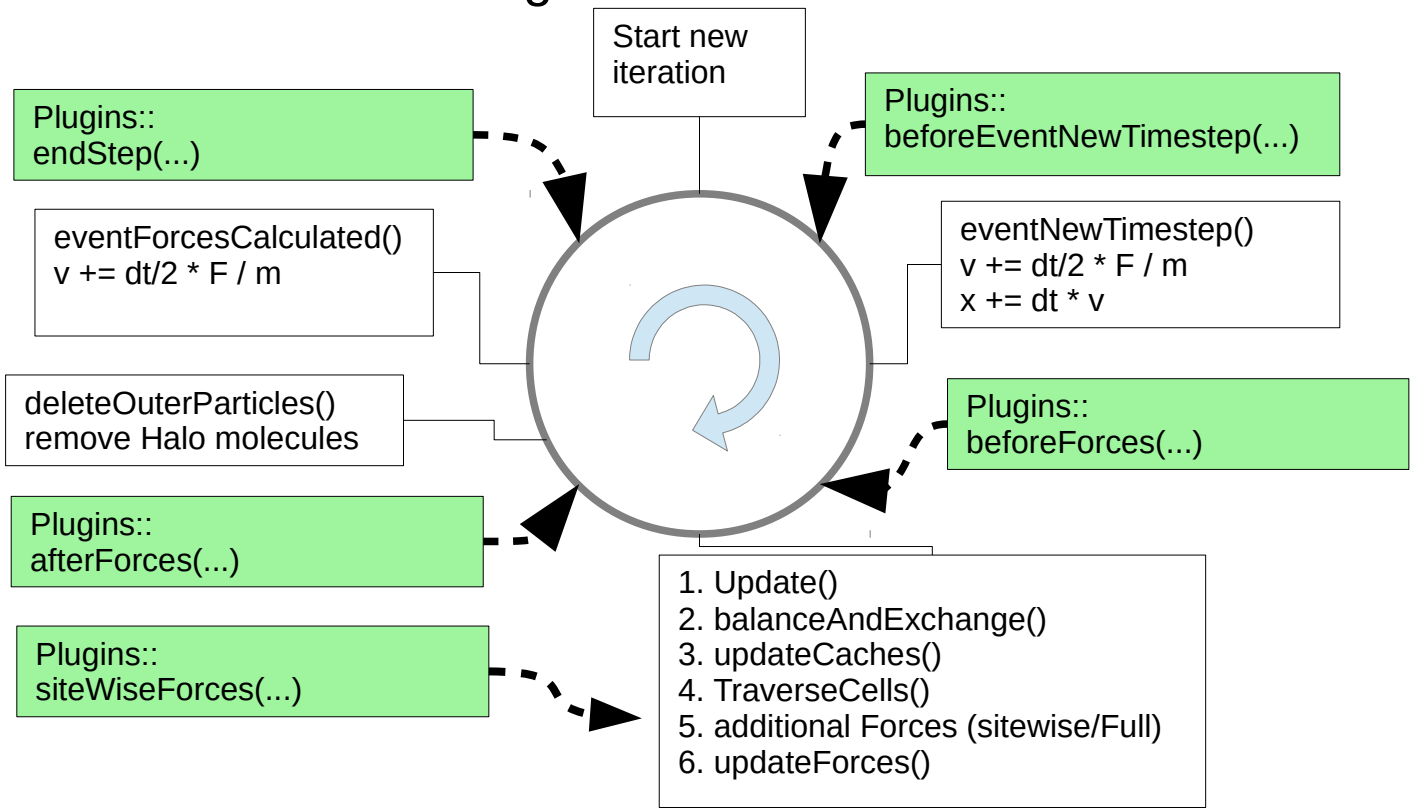
5) Document your plugin in the plugin summary:

   REPOSITORY/doc/Plugins_Summary.dox

6) Write a unit test for your plugin (can also be done after 2)

Example - HaloparticleWriter

1) Check whether the plugin already exists! => ok
2) Derive new class from PluginBase. => which functions do we need?



Example - HaloparticleWriter

1) Check whether the plugin already exists! => ok
2) Derive new class from PluginBase. => which functions do we need?

```
class HaloParticleWriter : public PluginBase {
public:
     HaloParticleWriter() {}
     ~HaloParticleWriter() {}

     void readXML(XMLfileUnits& xmlconfig);

     void afterForces(ParticleContainer* particleContainer,
               DomainDecompBase* domainDecomp, unsigned long simstep) override;

     std::string getPluginName() {
          return std::string("HaloParticleWriter");
     }
     static PluginBase* createInstance() { return new HaloParticleWriter(); }

};
```

Example - HaloparticleWriter

1) Check whether the plugin already exists! => ok
2) Derive new class from PluginBase. => ok
3) Write code.

```
void HaloParticleWriter::afterForces(ParticleContainer* pC, DomainDecompBase* domainDecomp,
                                     unsigned long simstep){
     if( simstep % _writeFrequency != 0 ) return;
     stringstream filenamestream;
     double rmin[3],rmax[3];
     filenamestream << _outputPrefix << "-rank" << domainDecomp->getRank() << ".halos.dat";
     std::ofstream checkpointfilestream(filenamestream.str());
     domainDecomp->getBoundingBoxMinMax(global_simulation->getDomain(), rmin, rmax);

     for (auto iter = pC->iterator(); iter.isValid(); ++iter) {
          if (not iter->inBox(rmin, rmax)) {
               checkpointfilestream <<"cell "<< tempMolecule.getCellIndex() << ": ";
               tempMolecule->write(checkpointfilestream);
          }
     }
     checkpointfilestream.close();
}
```

Example - HaloparticleWriter

1) Check whether the plugin already exists! => ok
2) Derive new class from PluginBase. => ok
3) Write code. => ok
4) Register plugin with PluginFactory.

```
#include "io/HaloParticleWriter.h"

template<>
void PluginFactory<PluginBase>::registerDefaultPlugins(){

...
    REGISTER_PLUGIN(GammaWriter);
    REGISTER_PLUGIN(HaloParticleWriter);
    REGISTER_PLUGIN(InMemoryCheckpointing);
...
}
```

Example - HaloparticleWriter

1) Check whether the plugin already exists! => ok
2) Derive new class from PluginBase. => ok
3) Write code. => ok
4) Register plugin with PluginFactory. => ok
5) Documentation (doxygen + summary page) => open
6) Unit Test => open

Further Reading

- Description of how to procede with plugins:
  https://www5.in.tum.de/mardyn/doxygen_doc/html/ls1GeneralPlugins.html
- (incomplete) List of Plugins:
  https://www5.in.tum.de/mardyn/doxygen_doc/html/PluginsSummary.html
  edit it here:
  REPOSITORY/doc/Plugins_Summary.dox
-