Open source engineering and sustainability tools for the built environment.
White Lioness technologies

http://www.white-lioness.com

http://www.bemnext.org

TU Delft
Delft University of Technology
Away from the clutter
1. Quality of the notebook to catch concepts

Requirements for future design systems
2. Process to follow design from start to end

Requirements for future design systems
3. Scalability

Requirements for future design systems
4. Flexibility

Requirements for future design systems
5. Openness

Requirements for future design systems
Open source?

Why bother?
Opportunity to learn

Open source code allows people to learn from what you have build.
To inspire and innovate!

Open source code allows people to become inspired by other people’s code
Opportunity to reuse

Open source code allows other people to reuse which means less waste and more intelligence.

We need to stop reinventing the wheel...
Opportunity to stand on the shoulders of giants

Open source code allows people to use other people’s specialist code so that not everybody has to reimplement difficult code.
Opportunity to reproduce and validate

Open source code allows other people to reproduce test data, validate the code and make it less vulnerable to errors...

And is the scientific system not founded on these values?
Don’t be evil

*Who really wants to belong to the dark side?*
Open source ≠ free

But sometimes it can be...
engineering-open

Open-source code for engineers
engineering-open
Open-source code for engineers

sustainability-open
Open-source tools for sustainable design
engineering-open
Open-source code for engineers
Yes, it is code...
Example
Example

spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)
Example

spacing=1.0
steel=Material(“steel”,2.1e5,0,0)
section=BeamSection2d(“HE200A”,steel)
structure=Structure(“001”)

DMSB 2013, Berlin, Germany, 2013-09-30
Example

```python
spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)
structure=Structure("001")
for i in range(10):
```

DMSB 2013, Berlin, Germany, 2013-09-30
Example

```python
spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
```
Example

```
spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")

for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))

for i in range(9):
```

DMSB 2013, Berlin, Germany, 2013-09-30
Example

spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Beam2d("beam", structure.n[i], structure.n[i+1], section))
Example

spacing = 1.0
steel = Material("steel", 2.1e5, 0.0)
section = BeamSection2d("HE200A", steel)

structure = Structure("001")
for i in range(10):
    structure.addNode(Node2d("node", (spacing*i), 0.0))
for i in range(9):
    structure.addElement(Beam2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
Example

spacing=1.0
steelm=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Beam2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
    structure.addLoad(Load2d("load", structure.n[i], 0.0, 1.0))
Example

spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Beam2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
    structure.addLoad(Load2d("load", structure.n[i], 0.0, 1.0))
structure.n[0].cx=True
Example

spacing=1.0
steel=Material("steel", 2.1e5, 0.0)
section=BeamSection2d("HE200A", steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node", (spacing*i), 0.0))
for i in range(9):
    structure.addElement(Beam2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
    structure.addLoad(Load2d("load", structure.n[i], 0.0, 1.0))
structure.n[0].cx=True
structure.n[0].cy=True
Example

spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Beam2d("beam",structure.n[i],structure.n[i+1],section))
for i in range(10):
    structure.addLoad(Load2d("load",structure.n[i],0.0,1.0))
structure.n[0].cx=True
structure.n[0].cy=True
structure.n[9].cx=True
Example

spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Beam2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
    structure.addLoad(Load2d("load", structure.n[i], 0.0, 1.0))
structure.n[0].cx=True
structure.n[0].cy=True
structure.n[9].cx=True
structure.n[9].cy=True
Example

spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Beam2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
    structure.addLoad(Load2d("load", structure.n[i], 0.0, 1.0))
structure.n[0].cx=True
structure.n[0].cy=True
structure.n[9].cx=True
structure.n[9].cy=True

fem2d=Fem2d(structure)
Example

spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Beam2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
    structure.addLoad(Load2d("load", structure.n[i], 0.0, 1.0))
structure.n[0].cx=True
structure.n[0].cy=True
structure.n[9].cx=True
structure.n[9].cy=True

fem2d=Fem2d(structure)
fem2d.analyse()
Example

```python
spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(Node2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Bar2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
    structure.addLoad(Load2d("load", structure.n[i], 0.0, 1.0))
structure.n[0].cx=True
structure.n[0].cy=True
structure.n[9].cx=True
structure.n[9].cy=True

dr=DynamicRelaxation(structure)
dr.analyse()
```

DMSB 2013, Berlin, Germany, 2013-09-30
Example

spacing=1.0
steel=Material("steel",2.1e5,0.0)
section=BeamSection2d("HE200A",steel)

structure=Structure("001")
for i in range(10):
    structure.addNode(MassedNode2d("node",(spacing*i),0.0))
for i in range(9):
    structure.addElement(Spring2d("beam", structure.n[i], structure.n[i+1], section))
for i in range(10):
    structure.addLoad(Load2d("load", structure.n[i], 0.0, 1.0))
structure.n[0].cx=True
structure.n[0].cy=True
structure.n[9].cx=True
structure.n[9].cy=True

pss=ParticleSpringSystem(structure)
pss.analyse()
Modules

Structure

Geometry
Modules

- Algorithms
- Structure
- Geometry
Modules

- Algorithms
- Calculations
- Structure
- Geometry
Implementation
Implementation

- Python
Implementation

- Python
- Can be used cross-platform (Windows, Mac, Linux)
Implementation

- Python
- Can be used cross-platform (Windows, Mac, Linux)
- Unit-testing
Implementation

- Python
- Can be used cross-platform (Windows, Mac, Linux)
- Unit-testing
- License: GPL v3
Implementation

- Python
- Can be used cross-platform (Windows, Mac, Linux)
- Unit-testing
- License: GPL v3
- Code is on Github:
Implementation

• Python
• Can be used cross-platform (Windows, Mac, Linux)
• Unit-testing
• License: GPL v3
• Code is on Github:
  http://github.com/jeroencoenders/engineering-open
Geometry
  VectorNd
  PointNd
  Point2d
  Point3d
### Geometry
- VectorNd
- PointNd
- Point2d
- Point3d

### Structure
- Material
- BeamSection2d
- Node2d
- Element2d
- Beam2d
- Load2d

#### Particle Spring
- MassedNode2d
- Spring2d
<table>
<thead>
<tr>
<th>Geometry</th>
<th>Structure</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>VectorNd</td>
<td>Structure</td>
<td>Particle Spring</td>
</tr>
<tr>
<td>PointNd</td>
<td>Material</td>
<td>FEM2d</td>
</tr>
<tr>
<td>Point2d</td>
<td>BeamSection2d</td>
<td>Dynamic Relaxation</td>
</tr>
<tr>
<td>Point3d</td>
<td>Node2d</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element2d</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beam2d</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Load2d</td>
<td></td>
</tr>
</tbody>
</table>

**Particle Spring**
- MassedNode2d
- Spring2d

---

**DMSB 2013, Berlin, Germany, 2013-09-30**
### Geometry
- VectorNd
- PointNd
- Point2d
- Point3d

### Structure
- Structure
- Material
- BeamSection2d
- Node2d
- Element2d
- Beam2d
- Load2d

### Algorithms
- Particle Spring
- FEM2d
- Dynamic Relaxation

### Calculations
- Simply Supported Beam

---

DMSB 2013, Berlin, Germany, 2013-09-30

White Lioness technologies
<table>
<thead>
<tr>
<th>Geometry</th>
<th>Structure</th>
<th>Algorithms</th>
<th>Calculations</th>
</tr>
</thead>
<tbody>
<tr>
<td>VectorNd</td>
<td>Structure</td>
<td>Particle Spring</td>
<td>Simply Supported Beam</td>
</tr>
<tr>
<td>PointNd</td>
<td>Material</td>
<td>FEM2d</td>
<td></td>
</tr>
<tr>
<td>Point2d</td>
<td>BeamSection2d</td>
<td>Dynamic Relaxation</td>
<td></td>
</tr>
<tr>
<td>Point3d</td>
<td>Node2d</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Element2d</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beam2d</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Load2d</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Some helper classes were left out of this list. For the full list, take a look at the code ;-)
Plans for the future

Geometry and math
- Meshes
- Vector operations

Form Finding algorithms
- Dynamic Relaxation
- Force Density Method

Optimisation algorithms
- Genetic Algorithms
- Simulated Annealing
- (B)ESO
Plans for the future

Geometry and math
• Meshes
• Vector operations

Form Finding algorithms
• Dynamic Relaxation
• Force Density Method

Optimisation algorithms
• Genetic Algorithms
• Simulated Annealing
• (B)ESO

Analysis
• 3D Finite Elements
• Bar elements
• Plate elements
• Shell elements
• Releases

Others
• Meshing algorithms
Plans for the future

Geometry and math
- Meshes
- Vector operations

Form Finding algorithms
- Dynamic Relaxation
- Force Density Method

Optimisation algorithms
- Genetic Algorithms
- Simulated Annealing
- (B)ESO

Analysis
- 3D Finite Elements
- Bar elements
- Plate elements
- Shell elements

Others
- Meshing algorithms

Insert your ideas/contributions <here>...

Who wants to do a GUI?
sustainability-open

“There should not be a reason anymore to not design in a sustainable manner”
sustainability-open

“There should not be a reason anymore to not design in a sustainable manner”
smart cities require smart systems

*The inter-connected world to come will be more complex to design, analyse and optimise with our current methods*
We need to measure first

To integrate and optimise we need to be able to measure first
paper = v0.0.1
presentation = v0.0.2
sustainability-open: original research project
dashboards
Tall Building Simulation
StructuralComponents
sustainability-open
sustainability-open framework

Grasshopper extensions
sustainability-open

myExtension

sustainability-open framework

myGHExtension

Grasshopper extensions
sustainability-open

myExtension

sustainability-open framework

myGHExtension

Grasshopper extensions

https://github.com/sustainability-open/sustainability-open#quick-example-of-a-framework-component
sustainability-open

https://github.com/sustainability-open/sustainability-open#quick-example-of-a-framework-component
sustainability-open
Website
http://www.sustainability-open.com
sustainability-open

Website
http://www.sustainability-open.com

Github
sustainability-open

Website
http://www.sustainability-open.com

Github
https://github.com/sustainability-open/sustainability-open
sustainability-open

Website
http://www.sustainability-open.com

Github
https://github.com/sustainability-open/sustainability-open
https://github.com/sustainability-open/so-bemnext-*
sustainability-open

Website
http://www.sustainability-open.com

Github
https://github.com/sustainability-open/sustainability-open
https://github.com/sustainability-open/so-bemnext-*
other websites...
sustainability-open v0.0.1

DMSB 2013, Berlin, Germany, 2013-09-30
Implementation

- Framework in C#.NET
- Since v0.0.2 largely independent from a modeling system.
- Implemented integration with Rhino/Grasshopper
The framework has an Apache 2.0 license.
Free to use and spread - even for commercial use.

The framework has an Apache 2.0 License.
Licensing

The framework has an Apache 2.0 license
Free to use and spread - even for commercial use
Components can have compatible licenses
Licensing

The framework has an Apache 2.0 license
Free to use and spread - even for commercial use
Components can have compatible licenses
The BEMNext Lab components will have another type of license
future developments [1/5]

Software integration
Parametric modeling systems
BIM systems
Other new design systems
future developments [2/5]

Design tools
Manual design, automated analysis and assessment
Automated design, analysis and assessment

Optimisation
“Engineer’s approach to design” (StructuralComponents)
- Metric-based
- Upper- and lower-boundary approaches
- Development for undefined to defined
- Continuous feedback
StructuralComponents
Design option exploration
sustainability-open

myExtension

sustainability-open framework

myGHExtension

Grasshopper extensions
### Future developments [3/5]

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building physics simulations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Behavioral simulation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other analysis types...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
future developments [4/5]

Assessment

Embodied energy breakdowns

Footprints

Operating energy breakdowns

Pollution and toxicity

Waste

Material depletion

Other impacts: noise, stench, vibrations, etc.

Opportunity assessment

Opportunities for recycling

Opportunities for upcycling

Demountability

Adaptability
future developments [5/5]

The uncertain (design) life-cycle
Continuous performance feedback during design
Influencing the design’s behaviour
Life-cycle analysis
Adaptability
Uncertain factors in the life cycle (how will the building be used?)
Human behaviour

Developments in systems, components and materials
Premium products
Energy production
Smart grids
Legolisation
The uncertain lifecycle

Total energy use

building life
The uncertain lifecycle

Total energy use vs. Energy investment in Design vs. Building life
The uncertain lifecycle
The uncertain lifecycle
The uncertain lifecycle

- Total energy use
- Investment in improved facade system
- Operational energy used during building life
- Embedded energy in construction
- Energy investment in design

DMSB 2013, Berlin, Germany, 2013-09-30
The uncertain lifecycle

- Energy Investment in Design
- Embodied Energy in Construction
- Energy Investment in Design
- Operational energy used during building life
- Reduced operational energy use
- Investment in improved facade system

Total energy use
The uncertain lifecycle

- Total energy use
- Operational energy used during building life
- Reduced operational energy use
- Investment in improved facade system
- Embodied Energy in Construction
The uncertain lifecycle

Energy Investment in Design

Embodied Energy in Construction

Operational energy used during building life

Reduced operational energy use

Investment in improved facade system

Investment in PVs

Total energy use

building life

Energy Investment in Design
The uncertain lifecycle

Total energy use

Embedded Energy in Construction

Energy Investment in Design

Operational energy used during building life

Reduced operational energy use

Investment in improved facade system

Investment in PVs

Net operational energy gain

Building life

Energy Investment in Design

DMSB 2013, Berlin, Germany, 2013-09-30
The uncertain lifecycle

- Energy Investment in Design
- Operational energy used during building life
- Reduced operational energy use
- Investment in improved facade system
- Total energy use
- Net operational energy gain
- Embodied Energy in Construction
- Energy Investment in Design
The uncertain lifecycle

- Energy Investment in Design
- Embodied Energy in Construction
- Energy Investment in Design
- Operational energy used during building life
- Reduced operational energy use
- Investment in improved facade system
- Investment in PVs
- Net operational energy gain
- Building as an energy producer
- Total energy use
- DMSB 2013, Berlin, Germany, 2013-09-30
The uncertain lifecycle

- **Energy Investment in Design**
- **Embodied Energy in Construction**
- **Operational energy used during building life**
- **Investment in improved facade system**
- **Investment in PVs**

### Operational Energy Use During Building Life
- **Investment in PVs**
- **Reduced operational energy use**
- **Net operational energy gain**

- **Building as an energy producer**
- **Total energy use**

---

DMSB 2013, Berlin, Germany, 2013-09-30
ok... and what do I do with this?
Dynamic analysis

Heat loss/gain ratio

DMSB 2013, Berlin, Germany, 2013-09-30
Do we still have time for a demo?
Thank you for your attention

@sustainable-open

http://www.white-lioness.com
@jeroencoenders