CHRONO::VEHICLE OVERVIEW

- Radu Serban
What is Chrono::Vehicle?

- **Chrono vertical app** (unit) for the modeling, simulation, and visualization of wheeled ground vehicles and (soon) tracked vehicles

- **Middleware**: can be embedded in third party software

- **Open source** with BSD license

- Library developed in **C++**

- **Cross-platform**: compiles on GNU GCC, MSVC, etc

- **Dependencies**: Chrono::Engine and (optionally) the Chrono UNIT_Irrlicht
What is Chrono::Vehicle?

- Vehicle system
- Auxiliary systems for testing vehicle systems in a co-simulation framework:
  - Tire system (rigid, LuGre, Pacejka)
  - Terrain (heigh-map)
  - Powertrain (engine + TC + transmission)
  - Driver model (interactive, data-based)
What is Chrono::Vehicle?

- **Modular**: vehicle are modeled from instances of subsystems (suspension, steering, driveline, etc.)
- **Flexible**: use parameterized templates
- **Expandable**, via C++ inheritance
  - New subsystems
  - New templates for existing subsystems
  - New vehicle types (e.g. tracked)
Code design – templates

- Template-based modeling
  (not in the C++ sense)

- In Chrono::Vehicle, templates are parameterized models that define a particular implementation of a subsystem type:
  - Define the basic Chrono modeling elements (bodies, joints, force elements, etc.)
  - Impose the subsystem topology (connectivity)
  - Define the template parameters
  - Implement common functionality for the type of subsystem (e.g. ‘suspension’), particularized to the specific template (e.g. ‘double-wishbone’)

5
Template example – Double wishbone suspension

- Upper control arm (UCA)
- Lower control arm (LCA)
- Ball joint (LCA balljoint, UCA balljoint)
- Shock absorber
- Tierod
- Upright
- Spindle
- Spindle revolute
- LCA revolute
- UCA revolute
- Lower control arm
DOUBLE A-ARM SUSPENSION TEMPLATE

- 3D rigid body
- 1D shaft element
- joint
- shaft – body connector

Chassis

Upright

Spindle

Axle

Angular velocity

Motor torque

Wheel state

Tire forces
DOUBLE A-ARM SUSPENSION

TEMPLATE PARAMETERS

- Mass and inertia tensor
- Rotational inertia
- Point location
- Spring coef., damping, free length

Diagram:
- Chassis
- Upper control arm
- Lower control arm
- Upright
- Spindle
- Axle

Joints:
- Revolute joint
- Spherical joint

Constraints:
- Distance constraint

Components:
- Shock
Code design – class hierarchy

- Chrono::Vehicle encapsulates templates for systems and subsystems in polymorphic C++ classes:
  - A base abstract class for the system/subsystem type (e.g. ChSuspension)
  - A derived, still abstract class for the system/subsystem template (e.g. ChDoubleWishbone)
  - Concrete class that particularize a given system/subsystem template (e.g. HMMWV_DoubleWishboneFront)

- Concrete classes:
  - User-defined – a derived class that satisfies all virtual functions imposed by the inherited template class
    - not part of the Chrono::Vehicle library
    - several example concrete classes and demo programs are provided
  - Generic – a derived class that satisfies all required virtual functions using parameter data from a specification file
    - part of the Chrono::Vehicle library
    - specification files use the JSON format
Chrono::Vehicle subsystem templates

- Suspension subsystem
  - Double wishbone
  - Multi-link
  - Solid-axle

- Steering subsystem
  - Pitman arm
  - Rack and pinion

- Driveline
  - 4WD
  - 2WD

- Brake

- Wheel
Run-time visualization with Irrlicht
Post-processing visualization with PovRay
Run-time and Presentation Animations
Code availability

- Source code: [https://github.com/projectchrono/chrono-vehicle](https://github.com/projectchrono/chrono-vehicle)
- Bug reports, feature requests, pull requests → GitHub
- API documentation (doxygen) [http://api.chrono-t.uwsbel.org/](http://api.chrono-t.uwsbel.org/)