# Composing PHP projects
# A new PHP package manager

Nils Adermann
naderman@phpbb.com
@naderman

August 23rd 2011

# Symfony2 Packages

Libraries / Components

Bundles

PSR-0 Autoloading

Git

# phpBB Packages

phpBB3

- MODX
- Extensions (>= 3.1)

phpBB4

- Regular Symfony2 Bundles
- “ExtensionBundles”

# NIH (Not Invented Here)

Wikipedias list of template engines:

- 33 PHP
- 15 Python
- 3 Ruby

Lists of >100 PHP engines on the web

Why is PHP at the top?

# Packages for End-Users

Web UI

browsing, installing, removing, updating

changes made in copy of vendor directory

A) overwrite old directory after update

B) download & upload vendor directory

C) (S)FTP support

# Packages for developers

Consistent versions across dev machines

Project specific package versions

Easy definition of dependencies in a file

Ability to work on dependencies' code

Simple definition of own packages

# PEAR / PEAR2 / Pyrus

Most people still use PHP4 PEAR

PEAR2 requires PHP 5.3

- uses namespaces

Pyrus is the PEAR2 Installer

- phar archive
- per-project package repositories

# Problems Pyrus does not solve

- Consistent versions across machines
- Web UI with vendor directory
  - hardcoded output, paths
  - directory transactions prevent self-update
- Working on dependencies' code
- Non-PHP dependencies

# Channels in PEAR

PEAR requires explicit channels

A channel cannot overwrite a package

You need to know channel locations

# Composer

Simple JSON declaration of packages

composer.lock file

Virtual Packages

Dependency Resolution with SAT

# Composer Repositories

- Anything
  - Git Repository, tar.gz, Local directory, ...
- Restriction to repository of approved packages possible
- Repositories can provide same packages
  - Prefered Repositories

# Packagist

http://packagist.org

Central repository of packages

All kinds of packages allowed

- Libraries
- Framework Bundles/Plugins/whatever
- Application Plugins (e.g. Drupal/Wordpress)

# SAT?

Boolean Satisfiability Problem:

Can `(A|B)&(C|D|(E&(A|D))&E` be true?

D(isjunctive)NF:

`(A&B)|(C&D&A)|(B&D)|C`

C(onjunctive)NF:

`(A|B)&(C|D|A)&(B|D)&C`

# A few numbers

100^2 = 10,000

100^3 = 1,000,000

2^100 ~=
1,267,650,600,000,000,000,000,000,000

# NP?

"hard" problems

"nondeterministic polynomial time"

no known polynomial time algorithm which:

- can solve all instances of the problem
- does not need to try all combinations

SAT is NP-complete (Cook's theorem)

# Package Management

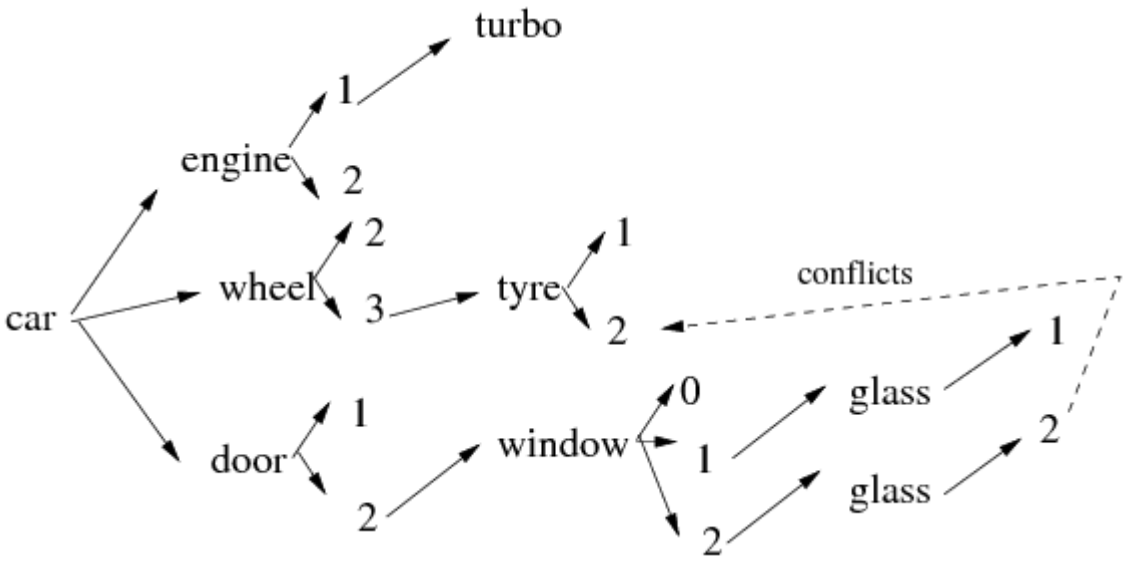
- Pool
- Repositories
- Package
  - Name & Version
  - requires, conflicts, provides

# Dependency Resolution

User wants to:

- install some packages
- update some packages
- remove some packages
- keep some packages

# Dependency Resolution

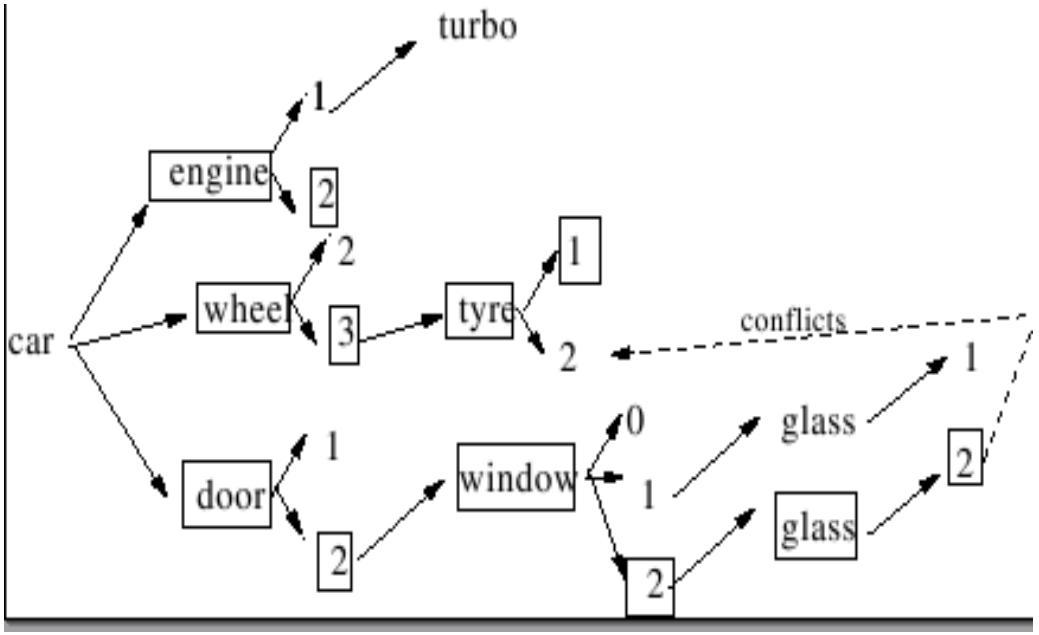



EDOS Project Workpackage 2 Team. Report on formal management of software dependencies. EDOS Project Deliverable Work Package 2, Deliverable 2, March 2006. http://www.edos-project.org/xwiki/bin/Main/Deliverables.

phpBB™ creating communities @ NEW YORK PHP

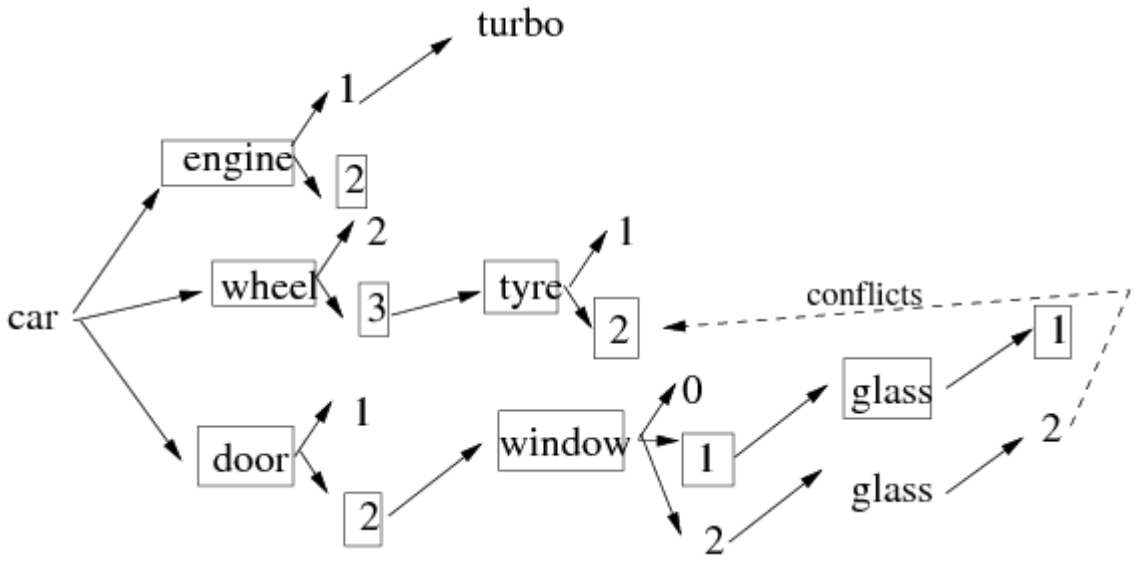
# Dependency Resolution

Ideal:



# Dependency Resolution

Acceptable:



# Dependency Resolution

No solution:

- APT
- Portage

Poor solution:

- smart
- Urpmi

# Dependency Resolution with SAT

SUSE's libzypper (Michael Schroeder)

Goals:

- Fast
- Complete & Reliable
- Understandable messages & explanations

Based on ideas from minisat

# Rule Generation

Job Rules:

Install A: `(A)`

Remove A: `(-A)`

# Rule Generation

Dependency Rules:

- A requires B: `(-A|B1|B2)`
- A conflicts with B: `(-A|-B1), (-A|-B2)`
- C and D provide A: A requires C or D

Policy Rules:

- B1 can be updated to B2: `(B1|B2)`
- B2 obsoletes B1: `(-B1|-B2)`

# Solving Rules

Unit Propagation:

```
(A|B|C), (-C), (-A|C)
  (-C)          C: false
  (-A|C)        A: false (-A: true)
  (A|B|C)       B: true
```

# Rule Solving Algorithm

- Unit Propagation
  - Contradiction?
    - Add Rule -X or unsolvable
    - Backtrack to previous assignment
- Assign undecided variable X (free choice)
- Repeat until solved

# Free Choices

Keep installed packages installed

Do not install unecessary packages

Pick newest version

# Composer Demo

# Resources

Composer & Packagist repositories

https://github.com/composer

Packagist:

http://packagist.org/

http://packagist.org/about

http://packagist.org/about-composer

# Thank you!

and Michael Schröder at Novell for his slides
http://files.opensuse.org/opensuse/en/b/b9/Fosdem2008-solver.pdf

http://www.mancoosi.org/edos/manager/
http://en.opensuse.org/openSUSE:Libzypp_satsolver

# Questions?