

---

# Designing Multilingual Applications

## International PHP Conference – Spring Edition

Kore Nordmann <kore@qafoo.com>

May 30, 2011

# About me

---

- ▶ Degree in computer science



# About me

---

- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP



# About me

---

- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects



# About me

---

- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects

Co-founder of



# About me

---

- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects

Co-founder of



**We help people to create  
high quality PHP  
applications.**

# About me

---

- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects

Co-founder of



**We help people to create  
high quality PHP  
applications.**

<http://qafoo.com>

# Motivation

---

- ▶ Multilingual / Multicultural applications can be hard



# Motivation

---

- ▶ Multilingual / Multicultural applications can be hard
- ▶ Different types of applications:
  - ▶ Single fixed language

# Motivation

---

- ▶ Multilingual / Multicultural applications can be hard
- ▶ Different types of applications:
  - ▶ Single fixed language
  - ▶ Consistent language per installation

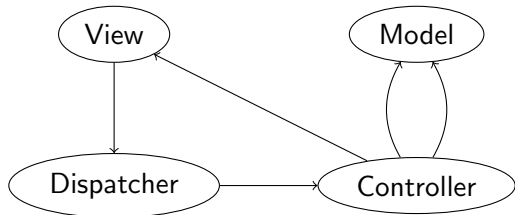
# Motivation

---

- ▶ Multilingual / Multicultural applications can be hard
- ▶ Different types of applications:
  - ▶ Single fixed language
  - ▶ Consistent language per installation
  - ▶ Multilingual content

# Where are the problems?

---



# Outline

---

The Dispatcher

The Model

The View

Conclusion

# Locate your user

---

- ▶ Parsing accept headers is not hard
  - ▶ Possible to do by regular expression

# Locate your user

---

- ▶ Parsing accept headers is not hard
  - ▶ Possible to do by regular expression
- ▶ *Never* use geolocation for language selection

# Locate your user

---

- ▶ Parsing accept headers is not hard
  - ▶ Possible to do by regular expression
- ▶ *Never* use geolocation for language selection
  - ▶ Or I'll get back at you!



# Locate your user

---

- ▶ Parsing accept headers is not hard
  - ▶ Possible to do by regular expression
- ▶ *Never* use geolocation for language selection
  - ▶ Or I'll get back at you!
- ▶ Result: `de` , `en_GB` , `cs_CZ.UTF-8` , `de_AU@euro` , ...
  - ▶ Format: `language[_territory][.codeset][@modifier]`

# Outline

---

The Dispatcher

**The Model**

The View

Conclusion

# The Model

---

- ▶ Translation of “user”-provided contents
  - ▶ News, Articles, Products, Comments, . . .

# Outline

---

## The Model

Model persistence

Domain-Problems

# One multilingual table

---

<b>id</b>	<b>language</b>	<b>state</b>	<b>title</b>	...
42	de	published	Dies ist ein Titel	
42	en	published	This is a title	

...

# One multilingual table

---

<b>id</b>	<b>language</b>	<b>state</b>	<b>title</b>	...
42	de	published	Dies ist ein Titel	
42	en	published	This is a title	
...				

- ▶ Dublication of non-translated items
- ▶ De-normalized

# Multiple columns

---

<b>id</b>	<b>state</b>	<b>title_1</b>	<b>title_2</b>	...
42	published	Dies ist ein Titel	This is a title	
...				

# Multiple columns

---

<b>id</b>	<b>state</b>	<b>title_1</b>	<b>title_2</b>	...
42	published	Dies ist ein Titel	This is a title	
...				

- ▶ Limited number of translations



# Multiple tables

---

id	state	...
42	published	

...

id	language	title	...
42	de	Dies ist ein Titel	
42	en	This is a title	

...

# Multiple tables

---

id	state	...
42	published	

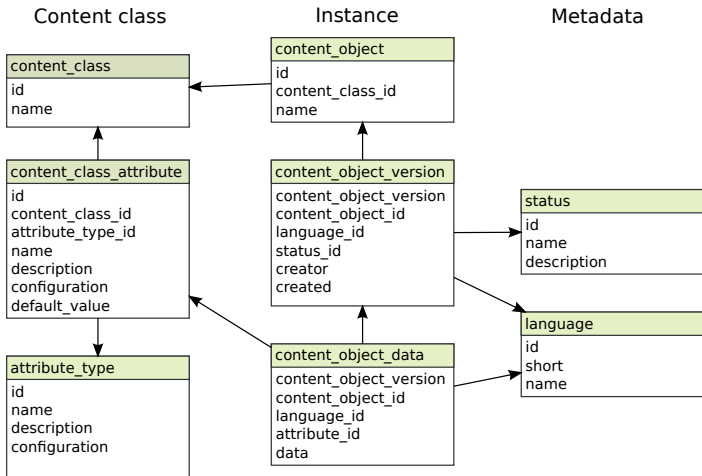
...

id	language	title	...
42	de	Dies ist ein Titel	
42	en	This is a title	

...

- More complex queries

# Full mapping



# Outline

---

## The Model

Model persistence

Domain-Problems

# The Model

---

- ▶ There is a bunch of domain-specific problems

# The Model

---

- ▶ There is a bunch of domain-specific problems
  - ▶ CMS
    - ▶ Different contents

# The Model

---

- ▶ There is a bunch of domain-specific problems
  - ▶ CMS
    - ▶ Different contents
  - ▶ Shop
    - ▶ Tax calculation
    - ▶ Legal requirements
    - ▶ Currency conversions

# The Model

---

- ▶ There is a bunch of domain-specific problems
  - ▶ CMS
    - ▶ Different contents
  - ▶ Shop
    - ▶ Tax calculation
    - ▶ Legal requirements
    - ▶ Currency conversions
  - ▶ Others
    - ▶ Recipe aggregation



# Content (CMS)

---

- ▶ Structure of content tree
  - ▶ Each item is translated into all languages

# Content (CMS)

---

- ▶ Structure of content tree
  - ▶ Each item is translated into all languages
  - ▶ Fallback language

# Content (CMS)

---

- ▶ Structure of content tree
  - ▶ Each item is translated into all languages
  - ▶ Fallback language
  - ▶ Display sub-tree for non-main languages

# Content (CMS)

---

- ▶ Structure of content tree
  - ▶ Each item is translated into all languages
  - ▶ Fallback language
  - ▶ Display sub-tree for non-main languages
  - ▶ Different contents per language

# Calculations (Shop)

---

- ▶ Rule based systems
  - ▶ Select calculation rules based on language / country

# Calculations (Shop)

---

- ▶ Rule based systems
  - ▶ Select calculation rules based on language / country
  - ▶ IF (@COUNTRY = "us" AND @STATE = "New York") @TAX = 23.43

# Calculations (Shop)

---

- ▶ Rule based systems

- ▶ Select calculation rules based on language / country
- ▶ IF (@COUNTRY = "us" AND @STATE = "New York") @TAX = 23.43
- ▶ IF (@COUNTRY = "de" AND @TYPE = "BOOK") @TAX = 7.00

# Calculations (Shop)

---

- ▶ Rule based systems

- ▶ Select calculation rules based on language / country
- ▶ IF (@COUNTRY = "us" AND @STATE = "New York") @TAX = 23.43
- ▶ IF (@COUNTRY = "de" AND @TYPE = "BOOK") @TAX = 7.00
- ▶ **DEFAULT: @TAX = 19.00**



# Outline

---

The Dispatcher

The Model

**The View**

Conclusion

# Translating strings

---

- ▶ `gettext()`
  - ▶ Depends on `setlocale()` , which is **not thread-safe**
- ▶ QT Linguist
  - ▶ XML based format used in QT
- ▶ PHP Arrays
- ▶ PHP Constants
- ▶ PHP implementation for `gettext` files
  - ▶ Should not depend on `setlocale()`
  - ▶ Will be slower

# Plural forms

---

- ▶ Most libraries somehow support plural forms
  - ▶ Support singular

# Plural forms

---

- ▶ Most libraries somehow support plural forms
  - ▶ Support singular
  - ▶ Support simple plural forms

# Plural forms

---

- ▶ Most libraries somehow support plural forms
  - ▶ Support singular
  - ▶ Support simple plural forms
  - ▶ Support different plural forms depending on number

# Plural forms

---

- ▶ Most libraries somehow support plural forms
  - ▶ Support singular
  - ▶ Support simple plural forms
  - ▶ Support different plural forms depending on number
  - ▶ Support multiple words with plural forms in one sentence

# Finding strings

---

- ▶ Locating untranslated strings
  - ▶ Plain strings are “trivial” to find

# Finding strings

---

- ▶ Locating untranslated strings
  - ▶ Plain strings are “trivial” to find
  - ▶ Parameterized strings are possible to find



# Finding strings

---

- ▶ Locating untranslated strings
  - ▶ Plain strings are “trivial” to find
  - ▶ Parameterized strings are possible to find
  - ▶ Error messages created in applications are impossible to find

# Finding strings

---

- ▶ Locating untranslated strings
  - ▶ Plain strings are “trivial” to find
  - ▶ Parameterized strings are possible to find
  - ▶ Error messages created in applications are impossible to find
- ▶ Handling untranslated strings
  - ▶ Use proper message as identifier (not MY\_FOO\_ERRORMSG1 )
    - ▶ Add “context”
  - ▶ Append strings to storage as “untranslated”

# Translations

---

- ▶ Standardized formats help Translators
  - ▶ Make it easy for *them*
  - ▶ There are various websites which make collaboration easy
    - ▶ Commonly support QT Linguist and gettext
  - ▶ There is a GUI for QT Linguist translations

# Example: crowdin.net

A lot of tools to manage community, translations quality, workflow, translation strategy...

Source strings that should be translated

Vote, Validate, Manage previous translations. Use the best one

Use machine translations and translation memory for reference

Collaborate by consenting translations

Powered by Crowdin.net

► <http://crowdin.net>

# Example: QT3 Linguist

The screenshot shows the Qt Linguist application window. The title bar reads "Qt Linguist by Trolltech - /home/kore/devel/arbit/trunk/src/translations/de.ts". The menu bar includes "File", "Edit", "Translation", "Validation", "Phrases", "View", and "Help". The toolbar contains various icons for file operations and editing.

The main interface is divided into three sections:

- Left Panel:** A list of translation items with columns for "Done", "Context", and "Items". The "modules/tracker" item is selected, showing 72/72 items done.
- Top Panel:** A table with columns "Done", "Source text", and "Translation". It lists items like "Open issues scheduled for %version", "Status", "Closed issues %closed / %issue\_count (%percent%)", "Issue listing", and "Issue actions".
- Bottom Panel:** A section titled "Phrases and guesses:" containing a table with columns "Source phrase", "Translation", and "Definition". It lists several phrases with their translations and associated keyboard shortcuts (e.g., "Guess (Ctrl+1)").

At the bottom right of the window, there is a status bar showing "126/126" and "MOD".

# Outline

---

The View  
Data handling

# Number formatting

---

```
1 <?php
2
3 var_dump( number_format( 42000.23, 2 ) );
4 // string( 9) "42,000.23"
5
6 var_dump( number_format( 42000.23, 2, ',', '.' ) );
7 // string( 9) "42.000,23"
```

# Number formatting

---

```
1 <?php
2
3 var_dump( number_format( 42000.23, 2 ) );
4 // string( 9) "42,000.23"
5
6 var_dump( number_format( 42000.23, 2, ',', '.' ) );
7 // string( 9) "42.000,23"
```



# Number formatting

---

```
1 <?php
2
3 var_dump( number_format( 42000.23, 2 ) );
4 // string( 9) "42,000.23"
5
6 var_dump( number_format( 42000.23, 2, ',', '.' ) );
7 // string( 9) "42.000,23"
```

# Number formatting

---

```
1 <?php
2
3 var_dump( number_format( 42000.23, 2 ) );
4 // string( 9) "42,000.23"
5
6 var_dump( number_format( 42000.23, 2, ',', '.' ) );
7 // string( 9) "42.000,23"
```

# Number formatting (Intl)

---

```
1 <?php
2
3 $formatter = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
4 var_dump( $formatter->format( 42000.23 ) );
5 // string(9) "42.000,23"
6
7 $formatter = new NumberFormatter( 'en_US', NumberFormatter::SPELLOUT );
8 var_dump( $formatter->format( 42000.23 ) );
9 // string(34) "forty-two thousand point two three"
```

# Number formatting (Intl)

---

```
1 <?php
2
3 $formatter = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
4 var_dump( $formatter->format( 42000.23 ) );
5 // string(9) "42.000,23"
6
7 $formatter = new NumberFormatter( 'en_US', NumberFormatter::SPELLOUT );
8 var_dump( $formatter->format( 42000.23 ) );
9 // string(34) "forty-two thousand point two three"
```

# Number formatting (Intl)

---

```
1 <?php
2
3 $formatter = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
4 var_dump( $formatter->format( 42000.23 ) );
5 // string(9) "42.000,23"
6
7 $formatter = new NumberFormatter( 'en_US', NumberFormatter::SPELLOUT );
8 var_dump( $formatter->format( 42000.23 ) );
9 // string(34) "forty-two thousand point two three"
```

# Number formatting (Intl)

---

```
1 <?php
2
3 $formatter = new NumberFormatter( 'de_DE', NumberFormatter::DECIMAL );
4 var_dump( $formatter->format( 42000.23 ) );
5 // string(9) "42.000,23"
6
7 $formatter = new NumberFormatter( 'en_US', NumberFormatter::SPELLOUT );
8 var_dump( $formatter->format( 42000.23 ) );
9 // string(34) "forty-two thousand point two three"
```

# Date formatting

---

```
1 <?php
2
3 $time = 1234567890;
4
5 var_dump( strftime( '%A, %e. %B. %G - %I:%M:%P', $time ) );
6 // string(38) "Saturday, 14. February 2009 - 12:31 am"
7
8 setlocale( LC_TIME, 'de', 'de_DE', 'de_DE.UTF-8', 'deu' );
9 var_dump( strftime( '%A, %e. %B. %G - %H:%M', $time ) );
10 // string(33) "Samstag, 14. Februar 2009 - 00:31"
```

# Date formatting

---

```
1 <?php
2
3 $time = 1234567890;
4
5 var_dump( strftime( '%A, %e. %B. %G - %I:%M:%P', $time ) );
6 // string(38) "Saturday, 14. February 2009 - 12:31 am"
7
8 setlocale( LC_TIME, 'de', 'de_DE', 'de_DE.UTF-8', 'deu' );
9 var_dump( strftime( '%A, %e. %B. %G - %H:%M', $time ) );
10 // string(33) "Samstag, 14. Februar 2009 - 00:31"
```



# Date formatting

---

```
1 <?php
2
3 $time = 1234567890;
4
5 var_dump( strftime( '%A, %e. %B. %G - %I:%M.%P', $time ) );
6 // string(38) "Saturday, 14. February 2009 - 12:31 am"
7
8 setlocale( LC_TIME, 'de', 'de_DE', 'de_DE.UTF-8', 'deu' );
9 var_dump( strftime( '%A, %e. %B. %G - %H:%M', $time ) );
10 // string(33) "Samstag, 14. Februar 2009 - 00:31"
```

# Date formatting

---

```
1 <?php
2
3 $time = 1234567890;
4
5 var_dump( strftime( '%A, %e. %B. %G. %I:%M.%P', $time ) );
6 // string(38) "Saturday, 14. February 2009 - 12:31 am"
7
8 setlocale( LC_TIME, 'de', 'de_DE', 'de_DE.UTF-8', 'deu' );
9 var_dump( strftime( '%A, %e. %B. %G. %H:%M', $time ) );
10 // string(33) "Samstag, 14. Februar 2009 - 00:31"
```

# Date formatting (Intl)

---

```
1 <?php
2
3 $time = 1234567890;
4
5 $formatter = new IntlDateFormatter(
6     'en',
7     IntlDateFormatter::FULL, // date
8     IntlDateFormatter::SHORT // time
9 );
10 var_dump( $formatter->format( $time ) );
11 // string(36) "Saturday, February 14, 2009 12:31 AM"
12
13 $formatter = new IntlDateFormatter(
14     'de_DE',
15     IntlDateFormatter::FULL, // date
16     IntlDateFormatter::SHORT // time
17 );
18 var_dump( $formatter->format( $time ) );
19 // string(31) "Samstag, 14. Februar 2009 00:31"
```

# Date formatting (Intl)

---

```
1 <?php
2
3 $time = 1234567890;
4
5 $formatter = new IntlDateFormatter(
6     'en',
7     IntlDateFormatter::FULL, // date
8     IntlDateFormatter::SHORT // time
9 );
10 var_dump( $formatter->format( $time ) );
11 // string(36) "Saturday, February 14, 2009 12:31 AM"
12
13 $formatter = new IntlDateFormatter(
14     'de_DE',
15     IntlDateFormatter::FULL, // date
16     IntlDateFormatter::SHORT // time
17 );
18 var_dump( $formatter->format( $time ) );
19 // string(31) "Samstag, 14. Februar 2009 00:31"
```

# Date formatting (Intl)

---

```
1 <?php
2
3 $time = 1234567890;
4
5 $formatter = new IntlDateFormatter(
6     'en',
7     IntlDateFormatter::FULL, // date
8     IntlDateFormatter::SHORT // time
9 );
10 var_dump( $formatter->format( $time ) );
11 // string(36) "Saturday, February 14, 2009 12:31 AM"
12
13 $formatter = new IntlDateFormatter(
14     'de_DE',
15     IntlDateFormatter::FULL, // date
16     IntlDateFormatter::SHORT // time
17 );
18 var_dump( $formatter->format( $time ) );
19 // string(31) "Samstag, 14. Februar 2009 00:31"
```

# Date formatting (Intl)

---

```
1 <?php
2
3 $time = 1234567890;
4
5 $formatter = new IntlDateFormatter(
6     'en',
7     IntlDateFormatter::FULL, // date
8     IntlDateFormatter::SHORT // time
9 );
10 var_dump( $formatter->format( $time ) );
11 // string(36) "Saturday, February 14, 2009 12:31 AM"
12
13 $formatter = new IntlDateFormatter(
14     'de_DE',
15     IntlDateFormatter::FULL, // date
16     IntlDateFormatter::SHORT // time
17 );
18 var_dump( $formatter->format( $time ) );
19 // string(31) "Samstag, 14. Februar 2009 00:31"
```

# Alternatives

---

- ▶ Zend\_Date
- ▶ Symfony
- ▶ arbitDateFormatter

# Intl gems

---

- ▶ Collator
- ▶ Normalizer
- ▶ Locale
- ▶ MessageFormatter
- ▶ NumberFormatter
- ▶ IntlDateFormatter



# Charsets & Encodings

---

- ▶ Use UTF-8

# Charsets & Encodings

---

- ▶ Use UTF-8
- ▶ Also see: [http://www.kore-nordmann.de/blog/php\\_charset\\_encoding\\_FAQ.html](http://www.kore-nordmann.de/blog/php_charset_encoding_FAQ.html)

# Outline

---

The Dispatcher

The Model

The View

**Conclusion**

# Conclusion

---

- ▶ There are really tricky domain specific problems
- ▶ Use solutions not depending on `setlocale()`
- ▶ Use intl extension if possible

# Thanks for listening

---

Please rate this talk at  
<http://joind.in/3491>  
and (optionally) give us some feedback right now

# Thanks for listening

---

Please rate this talk at  
<http://joind.in/3491>  
and (optionally) give us some feedback right now

**This is very important for ...**

- ▶ Speakers
- ▶ Organizers
- ▶ You!

# Thanks for listening

---

Please rate this talk at  
<http://joind.in/3491>  
and (optionally) give us some feedback right now

## Stay in touch

- ▶ Kore Nordmann
- ▶ [kore@qafoo.com](mailto:kore@qafoo.com)
- ▶ @koredn / @qafoo

**Rent a PHP quality expert:**

<http://qafoo.com>