SCG Pier migration

Ivan Ginovski
Mentor: Prof. Oscar Nierstrasz
Goal
Prerequisites

- knowledge of the old platform
- knowledge of the new platform (Drupal)
- knowledge of the problem domain
- knowledge of the migration process
Migration Strategy

Crawl website → Export → Import → QA → Finalize
Migration approaches

1. Naive approach
2. Semantic approach
3. Structural approach
Naive approach

Crawl the websites, extract content as a page and import into target CMS
Naive approach

**Pros:**
- Everything can be done externally
- Parse HTML

**Cons:**
- Which pages are dynamic or static
- Hardcoded HTML (no additional info)
- No control over CSS
- Resources (images/server files) – external retrieval, access problems
Semantic approach

Crawl the websites, extract content into elements and import into target CMS
Semantic approach

Pros:
- Optimal structure
- Everything is related

Cons:
- Resembles as a “starting work from scratch” project
- A lot more effort needed
- Difficult to identify entities
Structural Approach

Extract content from PierCMS into JSON data and import into target CMS
Structural Approach

Pros:
- Appropriate format in the target site
- Generic solution
- Flexibility to manage the content easier
PierCMS Content

- PRHeading
  - `<h1>Heading</h1>`

- PRLink
  - `<a href="/link">Title</a>`

- PRLList
  - `<ul><li>1. </li></ul>`

- PRTText
  - `<p>Text</p>`

- PRFormats
  - `<b>Bold format</b>`
Visitor Design Pattern

Visitor Design Pattern

Object → Visit: Visitor → Inspect

Accept:
Visitor Design Pattern

• Two essential functions:
  1. **accept: Visitor** v
     • Gives access of the object properties to the visitor
  2. **visit: Component** c
     • Performs operations on the object that is being visited (generate JSON)
Export Workflow

PR Content

Visit and write to stream

JSON Content

PRContent

The PRContent consist of PRPages and the components inside the page, such as Paragraph, Header, Links, Text, Style Formats etc.
Import Workflow

1. JSON Content
2. PHP SCRIPT
3. Pagedesigner Content

Drupal
Overview

Software Composition Group

The Software Composition Group carries out research in programming language design and software reengineering with the goal of building tools and development platforms for building, open-source software systems. The SCS is led by Oscar Nierstrasz and part of the Institute of Computer Science (ICS) at the University of Bern.

Research
- Agile Software Assistance

Teaching
Ongoing:
- SC: Software Composition Seminar

Autumn Semester 2020
- EI: Einschneidung in die Informatik (Bachelor)
- ESE: Introduction to Software Engineering (Einführung in IE) (Bachelor)
- SMA: Software Modeling and Analysis (Bachelor)
- Software Skills (Master)

Spring Semester 2021
- P3: Programming 2 — Object-Oriented Design (Bachelor)
- PL: Programming Languages (Bachelor)

Autumn Semester 2021
- EI: Einschneidung in die Informatik (Bachelor)
- ESE: Introduction to Software Engineering (Einführung in IE) (Bachelor)
- CP: Concurrency, State Machines and Design Patterns (Master)
- Software Skills (Master)

Autumn Semester 2020
- EI: Einschneidung in die Informatik (Bachelor)
- ESE: Introduction to Software Engineering (Einführung in IE) (Bachelor)
- SMA: Software Modeling and Analysis (Bachelor)
- Software Skills (Master)

Spring Semester 2021
- P3: Programming 2 — Object-Oriented Design (Bachelor)
- PL: Programming Languages (Bachelor)

Autumn Semester 2021
- EI: Einschneidung in die Informatik (Bachelor)
- ESE: Introduction to Software Engineering (Einführung in IE) (Bachelor)
- SMA: Software Modeling and Analysis (Bachelor)
- Software Skills (Master)

http://scg.unibe.ch
https://ig-drpl.docker-dev.iqual.ch

Static webpages migrate demo
Easier to do manually:
TODO: URL Paths

- https://scg.unibe.ch/path/to/page
- https://ig-drpl.ch/node/23
- “/path/to/page” included in JSON
- Solution: Pathauto Module https://ig-drpl.ch/path/to/page
TODO: News Block, SCGBib

Software Composition Group

The Software Composition Group carries out research in programming language design and software reengineering with the goal of facilitating the development of flexible, open software systems. The SCG is led by Oscar Nierstrasz and is part of the Institute of Computer Science (IFW) at the University of Berne.

Research
- Agile Software Assistance

Teaching
Ongoing:
- SC: Software Composition Seminar

Autumn Semester 2020
- EI: Einführung In die Informatik (Bachelors)
- ESE: Introduction to Software Engineering (Einführung in SE) (Bachelors)
- SMA: Software Modeling and Analysis (Masters)
- Software Skills (Masters)

Spring Semester 2021
- P2: Programming 2 — Object-Oriented Design (Bachelors)
- PL: Programming Languages (Masters)

Autumn Semester 2021
- EI: Einführung In die Informatik (Bachelors)
- ESE: Introduction to Software Engineering (Einführung in SE) (Bachelors)
- CP: Concurrency: State Models and Design Patterns (Masters)
- Software Skills (Masters)

Highlight

Pharo by Example is an open-source book about the Pharo development environment for the classic Smalltalk-80 programming language.

News

Follow scg news on twitter

Modifiable requirements
12 December 2020
Nithin Patkar: Modifiable requirements. In Benavi’20, p… . 2020 Details. Abstract Separate tools a…

A Sampling Profiler for a JIT Compiler
4 September 2020
Andreas Wältli: A Sampling Profiler for a JIT Compiler. Masters thesis, University of Berne, Se…

Modeling requirements artifacts in an IDE
1 September 2020

Search SCG Bibliography


Summary

Structural Approach

- Static pages
- URL Paths
- News block
- SCGBib
- Dynamic pages
Thank you for your attention
Backup slides with technical details
Naive approach

Crawl the websites, download content and import into target CMS.
Semantic approach

Pros:
• Optimal structure
• Everything is related

Cons:
• Resembles as a “starting work from scratch” project
• A lot more effort needed
• Difficult to identify entities
Visitor Design Pattern

• Visitor is a behavioral design pattern.
• Represents an operation to be performed on the elements of an object structure.
• Visitor lets you define a new operation without changing the classes of the elements on which it operates.
Migration Strategy

• Phase I: Implementing program of web crawling and webpage parsing => output
• Phase II: Implementing program to import JSON data into Drupal 8 site (PHP)
• Phase III: QA test
• Phase IV: Finally, to run programs to migrate the whole site to Drupal 8 site on Production.
Migration Steps

- **Crawl** – List of all the website URLs (XML sitemap)
- **Analyze CMS Features**
  - Widgets ➔ Plugins
- **User Stories and Acceptance Criteria**
- **Final Checks**
  - Search for errors/bugs/broken URLs
- **Monitor traffic and Post-Migration fixes**

Keep or leave out?,
Do 1 picture with these two slides
Naive approach

• 1. Crawl the existing website and download all the content
• 2. Create the templates needed for the content
• 3. Import the HTML contents into the new CMS
Structured Approach

• 1. Export all the content into general structure (JSON)
• 2. Target CMS that has functionality for the features (news/scg bib filtering/pdf generate pages/cronjob)
• 3. Create the templates needed for the content
• 4. Import the content to the CMS (in the appropriate format)
   - PHP Script - logic that can create the content from the structured data
Semantic Approach

• 1. Export all the content into general structure (JSON)
• 2. Export/Convert raw data in Pier into structured data (e.g. teachings, classes, dates, PDFs)
• 3. Import the content to the CMS
  - logic that can create the content from the structured data
Structured Approach

• Pros:
  • Appropriate format in the target site
  • Generic solution
  • Flexibility to manage the content easier

• Cons:
Semantic Approach

• Pros:
  • Optimal structure
  • Everything is relative and connected – more robust

• Cons:
  • Resembles as a “starting work from scratch” project
  • A lot more effort needed
  • Multiple iterations needed for referencing each node (content entity)