



***Non-hierarchical categorization
How it can work?***

Patrik Fältström
Tele2/Swipnet
paf@swip.net

Content

- Categorization
 - What is it?
 - What is it used for?
- Classical categorization
 - One view, one viewer
- Non-hierarchical categorization
 - It works!
- Summary

What is categorization

- It makes it easier to find information
 - I want to find information about Visby in the 12th century!
- It makes people understand what the material is about
 - Is this book named “Rock” about music or geology?
- On the web, it is used for both purposes

Brief discussion about Metadata

- Metadata is in the normal world created by “librarians”
- On the net, the author is often creating the metadata
 - ...and we have no other choice for now
- We all know that the author is not the best choice!

The author as a classifier

- Who is the author?
 - He probably knows the topic :-)
 - He does not know about similar work
 - He does not know about other work
 - He doesn't know about any classification scheme
- He need some help!
 - The classification is for the audience, not the author!

The audience

- Uses the categorization
 - To find the work
 - To find similar work
 - Uses classification and abstract to know if this is some interesting work
- Different users have different knowledge
 - I will get back on this topic...



Food / Recipes

- We have starters, main courses and desserts
- From different parts of the world
- With different main ingredients

Food (more...)

- I want something Thai...
 - Food -> Asian -> Thai
- I want a chicken main course...
 - Food -> Chicken -> Main Course
 - Food -> Main Course -> Bird -> Chicken
- We need different views!

So, what's the point?

- Categorize with “tokens” only
 - Thai, Asian, Food, Chicken, Bird, Main Course
- Rules which state what goes where in the hierarchy

Main node

- Name: Main
 - The name of this node
- Rule: <.>
 - One item only
- Attributes: <>
 - Empty string

Food node

- Name: Food
- Rule: <food,.>
 - “food” and one more item
- Attributes: <Food>
 - One string only, so this node ends up on the main menu

Main Course Node

- Name: Main Courses
- Rule: <Food, Main course, .>
- Attributes: <Food, Main Course>

Chicken Main Course Node

- Name: Chicken Main Courses
- Rule: <Main Course, Chicken, *>
 - Note the star, which means that all nodes with these two items and other aswell (any number of them) will show up in this node
- Attributes: <Food, Main Course, Chicken>

Thai node

- Name: Thai dishes
- Rule: <Food, Thai, *>
 - Also a star, which is a wildcard
- Attributes: <Food, Asian, Thai>

Kai Tod Taucheo

- **Name:** Deep-Fried Chicken with Yellow Bean Sauce
- **Attributes:**
<Food, Thai, Chicken, Main Course>
- **Matches both:**
 - <Food, Thai, *>
 - <Main Course, Chicken, *>

Get the point?



Note that!

- The document is visible regardless if the hierarchy is changed or not
- The document is visible in multiple places in the hierarchy
- The person categorizing does not need to know the structure of the tree

This exists already!

- The software is developed by Tele2 and Bunyip Information Systems
 - Robot - Jackaroo
 - Database - Digger
 - Menu system - Koala
- Jackaroo also does automatic classification if Dublin Core elements are missing in the html pages

Where to get more information?

- My email address:
paf@swip.net
- Myself:
<http://paf.se>
- Bunyip Information Systems
<http://www.bunyip.com>
- Tele2
<http://www.tele2.se>