

Funkfeuer Node Database

Dr. Ralf Schlatterbeck
Open Source Consulting

Email: office@runtux.com
Web: <http://www.runtux.com>
Tel. +43/650/621 40 17



Funkfeuer Node Database

- Developed in a EU research project
- Implementation language Python
- Should replace current “Redeemer” application
- No monolithic approach
- Use **RESTful** API for applications instead
- Work with other community networks to develop a common API
- ... then we may finally re-use components



No Common Software

- Historically many Node DB implementations
- Every network invents their own
- But requirements differ a lot
- One solution good at monitoring (WLAN Slovenia)
- Others allow users to advertise own services (Athens)
- Some manage uplink centrally (Funkfeuer)
- In others users have to find someone for uplink (guifi.net)
- Some use **OLSR** for routing (Funkfeuer)
- Others use **BGP** for routing (guifi.net)



No Common Software

- Operators want to be able to change their code
- ... and stay in control
- ... that's part of what a community network is for
- but tastes of programming languages and used frameworks vary
- ... there might not be a single solution

→ Use a common API so that we can share apps
→ map-display, config-file generation, link planning ...



New Solution

- Yet another implementation
- With **RESTful** API
- And good templating engine (not just for HTML)
- Based on a new framework Tapyr
- which integrates best-of-breed tools
 - + SQLAlchemy as object-relational-mapper (ORM)
 - + Werkzeug WSGI utility library
 - + Jinja2 templates
 - + ...

→ Code at github.com/FFM/FFM



New Solution: Demo

- Documentation: ffm.gg32.com/Doc
- Admin Interface: ffm.gg32.com/Admin
- Login for Admin Interface available on request (Demo contains no personal Data)
- ffm.gg32.com/Admin/FFM/IP4_Network
- ffm.gg32.com/Admin/FFM/Antenna
- ffm.gg32.com/Admin/FFM/Regulatory_Permission



Framework Tapyr

Tapyr Framework consists of several parts

- MOM (meta object model) for data modelling uses **SQLAlchemy** (SQLAlchemy by Michael Bayer)
- GTW Web application and/or **RESTful** Server uses **Werkzeug** (First implementation based on **Tornado**)
- JNJ uses **Jinja2** for templating
- modular, parts are independent, use best-in-class components
- Jinja2 and Werkzeug are by Armin Ronacher
- Tapyr by Christian Tanzer and Martin Glück
- Code: github.com/Tapyr/tapyr



Framework Tapyr

- Modular: Use parts of the framework
 - you may use only MOM (e.g. conversion of Funkfeuer Redeemer data)
 - ... or you run only **RESTful** API without templating
 - ... or you write a QT app without web stuff but with MOM object model
- Comprehensive use of namespaces (packages)
 - should be short to avoid unreadable code
 - we typically use 3-letter names (on the top-level)



Framework Tapyr MOM: Essential modelling

- Essence: concept by Aristotle “attributes that make an entity or substance what it fundamentally is, and which it has by necessity, and without which it loses its identity. Essence is contrasted with accident: a property that the entity or substance has contingently, without which the substance can still retain its identity.” (Wikipedia: [Essence](#))
 - Excludes implementation artefacts (think of a perfect technology that makes accidental attributes unnecessary) [MP84]
- Include all essential attributes



Framework Tapyr MOM: Essential modelling

- Essential model describes application domain as a number of essential types and their relations
- Essential type characterized by
 - Attributes (definition includes all knowledge needed)
 - Essential primary key
 - Predicates
 - ... for a single or several attributes
 - ... for several entities



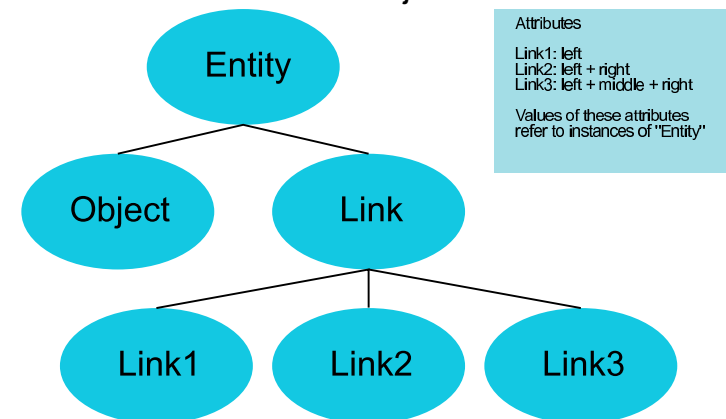
Framework Tapyr MOM: Essential modelling

- Model links as separate entity (essential type) [Tan95]
 - Link is a first class entity
 - Link to links (!)
 - Antenna is Link1 to Antenna_Type
 - Wireless_Interface_uses_Antenna is a link to a link
 - Polymorphic Links possible: link to a base-class (!)



Framework Tapyr MOM: Essential modelling

Base classes of MOM object model





Framework Tapyr MOM: Types, Attributes

Attribute definition includes all knowledge needed

- Introspection / Reflection: Info about types
 - Information about inheritance and associations
 - ID of an object is not type-specific
 - Polymorphic queries possible
 - ... strict: same class, non-strict: children
 - ... e.g. ask about all changes in the last week
 - Polymorphic links possible (!)
- generate browsable documentation
→ generate admin web-interface
→ generate generic **RESTful** API



Framework Tapyr MOM: Types, Attributes

- Attributes have a name and a type
- Inheritance for Attributes
- Optional verbose description
- Kind: Primary, Required, Necessary, Optional, ...
- Constraints: predicates on attribute
- Optional default value
- Mandatory example
- Attributes can be computed
- ... or can map to a database query
- Cooked values (use for computation) vs. Raw input
- String representation



Framework Tapyr MOM: Cooked / Raw

- User inputs raw value
- Cooked value used for computation → query
- Normally an attribute stores the cooked value
- But raw value can be stored in addition or instead
- ... e.g. for auto-completion
- Calendar date: store cooked value (date/time)
- Frequency value: store raw and cooked value (user may enter “2400 MHz” and see this on next edit)
- Case insensitive strings, cooked value is lowercase, store both
- Encoding: strings are unicode internally (cooked)



Framework Tapyr MOM: RESTful API

- ffm.gg32.com/api/FFM-Device
- ffm.gg32.com/api/FFM-Device?verbose
- ffm.gg32.com/api/FFM-Antenna?verbose
- ffm.gg32.com/api/FFM-Antenna?verbose&raw
- ffm.gg32.com/api/FFM-Antenna?verbose&ckd&raw
- ... [?verbose&AQ=gain,GT,20](http://ffm.gg32.com/api/FFM-Antenna?verbose&AQ=gain,GT,20)
- ... [?verbose&AQ=azimuth,GT,90&AQ=azimuth,LE,](http://ffm.gg32.com/api/FFM-Antenna?verbose&AQ=azimuth,GT,90&AQ=azimuth,LE)
- ... [?verbose&AQ=band.lower,GT,2800 MHz](http://ffm.gg32.com/api/FFM-Antenna?verbose&AQ=band.lower,GT,2800)
- Polymorphic query:
Antenna is_a Device, Net_Device is_a Device
ffm.gg32.com/api/FFM-Device?verbose



Framework Tapyr MOM: Backends

- MOM is Backend-independent
- Different relational databases (uses SQLAlchemy)
- Python Pickle store
- Easier migration via export to Pickle and back to (other) relational backend
- Also works for complex schema changes



Framework Tapyr GTW

GTW, why not use Werkzeug directly?

- Tree of resources
 - Tree of web pages
 - Tree of **RESTful** resources
- Supports aliases (which allows to deviate from tree structure)
- Define HTTP methods (get, put, post, ...) per resource
- Representations per resource (JSON, HTML, CSV, ...)
- Authorisation and Permissions



Framework Tapyr GTW

- Inheritance of Permissions and representations (down the tree)
- Good for auto-generated navigation hints
- Navigation: To next/prev Neighbor (e.g. for picture gallery)
- Other auto-generation features, e.g., robots.txt, site map
- Request processing
- URL dispatch
- Generated regular expressions for URL mapping



Framework Tapyr: Jinja2

- Good support for factorization and modularization
- Jinja2 macros correspond to python functions
- Extensible
- Better than Django (first try used Django)
- Explicit whitespace control without destroying template structure
- Fast (compiles templates to bytecode)
- Include vs. import
- Can call templating macros from python (AJAX!)
- Not HTML specific
- Pythonic :-)



Framework Tapyr: JNJ

- May define 1 .media file per .jnj template module
- Contains python generating css
- Styles are symbolic and can be configured
- Contains python referring to javascript
- Dependencies of .js files defined in python
- Use javascript and css minimizers that use transitive closure per script
- File with hash filename .css or .js for each resource one file, filename is hash over content
- Cache that maps resource to needed files



Bibliography

- [MP84] S. M. McMenamin and J. F. Palmer. *Essential Systems Analysis*. Yourdon Press, New York, 1984.
- [Tan95] Christian Tanzer. Remarks on object-oriented modelling of associations. *Journal of Object-Oriented Programming*, 7(9):43–46, February 1995.



Contents

Funkfeuer Node Database	2
No Common Software	3
New Solution	5
New Solution: Demo	6
Framework Tapyr	7
Framework Tapyr MOM: Essential modelling	9
Framework Tapyr MOM: Types, Attributes	13
Framework Tapyr MOM: Cooked / Raw	15
Framework Tapyr MOM: RESTful API	16
Framework Tapyr MOM: Backends	17



Contents

Framework Tapyr GTW	18
Framework Tapyr: Jinja2	20
Framework Tapyr: JNJ	21
Bibliography	22