Beings & Issues & **Passions** | Beings & Groups | Groups & Happenings | Happenings & Issues | Happenings & Beings | Issues & **Passions**

About 'InterEthos'

Overview | Stories | Data Model | Crows | Philosophy | Geek Notes

Overview

Create a system to allow users to create and apply taxonomy indexes (ie. a "folksonomy" system) against the human spirit, and apply those to actual people's profiles. At its core the data system must have the flexibility to allow users from different cultures to be able to express totally different ways of thinking about spirituality (ie. taxonomy structures).

Users will also need to be able to create "lenses" with which to view the endless potential outputs that such a system will provide. Different users will want different lenses to view people around them. Users will also be able to suggested lenses that visitors might use in viewing their profile.

Subsystem of three validation levels of user identities:

1. non- validated users: user identities created by parties other than the owner, and not claimed or maintained by the owners,
2. partially validated users: owners create and pro-actively link to their own identity, and
3. fully validated users: owners create, verify, and maintain their identity with a structured regularity and protocol.

Another dimension of validation will be the "longevity" of a person's identity within the system (ie. users who change their profiles "like people change their socks" will be given less authority than users who continually validate and maintain their profiles over many years). Higher validation levels will add authority to a user's postings to other users' profiles. This will be very important to protect against character assassination attempts.

Browser add-on application(s) will allow users to

- input a list of "people to watch for", which "listens" for appearances of those names on visited content.
- Browsers may be visiting websites where mention is made of people measured by the system, and the depicted the "ethos" of said people will be displayed in the add-on's toolbar.
- When such references are found, then the add-on application will pull and render interpretations of the XML profiles for referenced people (based on aforementioned "lenses")
- The same renderings will allow manipulation of (posting to) those people's profiles. (Posting to other people's profiles will require that the users be registered in the system.)

With the creation of a core data platform, then a system will be created to replicate the server into a "cloud" network of servers. The ultimate goal is to create a cloud that shard the overall data, and which preclude the possibility of one entity owning and/or controlling the platform.

One goal is for a person to be able to walk into a physical ballroom filled with a hundred people whom s/he has never met, and be able to immediately "assess" who is in the room. Maybe the person is looking for "like-minded" people, or people with similar skills, or people with a similar Myers Briggs profile. Maybe the person is looking for a "diamond in the rough" person who can satisfy a particular need in a current project.

Of course the same situation might happen in a virtual reality ballroom, similar to Hero's trip to the bar in the book "Snow Crash".

Here is a blog that I have maintained for the past year on this project:
https://interethos.wikispaces.com

Overview Stories Data Model Crows Philosophy Geek Notes

Preface

"Beings" can be part of "Groups", but they do not have to be.

"Issues" are concepts in life that have multiple correct answers, and your core beliefs pertaining to those Issues define portions of spirit. Things published by you, or about you, can help define your beliefs surrounding specific Issues. So can your answering certain simple 'litmus test' questions.

"Passions" is the label we used for those core beliefs of Beings. Passions are ultimately measured and shared, toward the end of harmonizing the people of the world.

"Groups" of Beings are invited to answer questions that are specific to, and crafted by the visionary leaders of, those groups. These questions aim to define Beings stance on the Issues associated with the Group.

"Happenings" are the scheduled events or ongoing initiatives of Groups.

"Realms" are linked to from the center of the footer toolbar, a different location from all other indices, as they apply to many of the other indices, and yet are not dependent in the object architecture.

Stories

Profile registration

1. Users can register system profiles for
2. themselves
3. other people, even without permissions of those people

4. Users can validate ownership of their profile by acknowledging the primary email address associated with it - ***Phase 2***
5. Users can elect to receive individual or "daily/weekly/monthly digest" notifications about posts affecting their profile - ***Phase 2***
6. Users can elect to monitor and maintain their profile
7. If they choose to monitor their profile, then auto-reminders will periodically be emailed to them reminding them - ***Phase 2***

Profile definition

1. Users must be registered with the system, and also must acknowledge any posts they make as depicted in the following bullets. - ***Phase 2***
2. Users can choose to receive a daily or weekly "digest" of their posts, which may be desirable for intensive users of the system. - ***Phase 2***
3. Users can add Realms to the taxonomy (ie. it is a user-created folksonomy)
    1. A new index must include
        1. Name - English names will be used to start, until language indexing functionality is created
        2. Parent - new index must be children of other pre-existing indexes. This assumes that all top-level indexes are set up at launch, and/or can be requested by users through a special "top-level index addition" process.
4. Users can create new Issues and Passion tag sets used to discern some aspect of human spirit, and which will consist of one of the following:
    1. A written question - TBD
    2. A URI pointing to an existing piece of content - TBD
5. Users can create new tags which must include
    1. An Issue
    2. Two Realms (or categories) indicating the two opposite elements of spirit that will be delineated by the URI or question.
6. Users can post tags (existing or new) to users - TBD
7. Users who wish to "rapidly create and post a tag" need a streamlined one-screen process. It is assumed that the User ID of the person being posted against will be supplied up front, to the first screen. So the user will be able to do the following all in a single screen.
    1. "Create a Way Point" and - TBD
    2. "Post to User". - TBD
8. Users can (be invited to) answer
    1. an individual question, or - TBD
    2. a group of questions - TBD

User groups

1. When a user is created it is not associated with any group, but afterwards may be associated with any number of groups.
2. Users can be emailed invitations to participate in a group, complete with "subject", "descriptive text", and "host contact information" of the invitation. - TBD
3. Groups allow multiple profiles to be viewed (see "Profile viewing" below) and compared with the download of a single "click". - TBD
4. People's profiles can be grouped as might happen in real life as follows: - TBD
    1. A convention of 250 people might benefit from having all members profiles downloadable with a single group download. (A perfect example of a group is

        the "ballroom filled with a hundred people" depicted in the "Conceptual Overview" section above. A person who is about to walk into the hall will have direct access to pull all of the profiles for every one in the group.)
    2. An ongoing "club" might benefit from having an active roster
5. Group accessibility can be set up to as follows:
    1. Accessible by the public
    2. Hidden from the public, and only accessible by group members - ***Phase 2***
    3. Administrators for a group must be "subscribed" to the group. - ***Phase 2***

Profile viewing

1. Users can view (pull):
    1. a single person's profile, or - TBD
    2. a group of users's profiles - TBD
2. The data will be served almost exclusively as XML data sets. - TBD
3. Users can create lenses describing how they want to view profile data in general for - ***Phase 2***
    1. All users.
    2. Certain segments of users
    3. Lenses can be specific to a group
4. Users need a viewer to render the XML data in a user-friendly format - ***Phase 2***
    1. Hand-held cell phones and iTouch-like devices need installable applications - TBD
    2. Graphical icons need to be created to "pictorial-ize" larger profile elements - TBD
    3. Web browsers will be used as a default viewer, but later custom applications will be built - ***Phase 2***

Profile dispute process - *Phase 2*

1. Users can dispute tag posts to any profile
    1. Such disputes will not remove or change a tag post
    2. Disputes will reflect record of their existence back to the profile of the user that posted the disputed tag.
2. Users will be encouraged to resolve all disputes, as disputes will reflect back on their own profile poorly
3. Users can invite other users to assess their disputed situation, and either support their dispute or validate the disputed tag that was posted against their profile.

Languages - *Phase 2*

All interface wording must be localized by language and dialect, including interface field names and directions, and all data points. For the sake of visual simplicity the "Languages" table has been removed from the data model graphic (below), as it links to almost all other tables.

Implementation Notes (for Phase 1 only)

Custom integrations to the normal Ruby on Rails environment

1. "Acts as Tree" add-on to render interactive menu trees for self-referential Realms

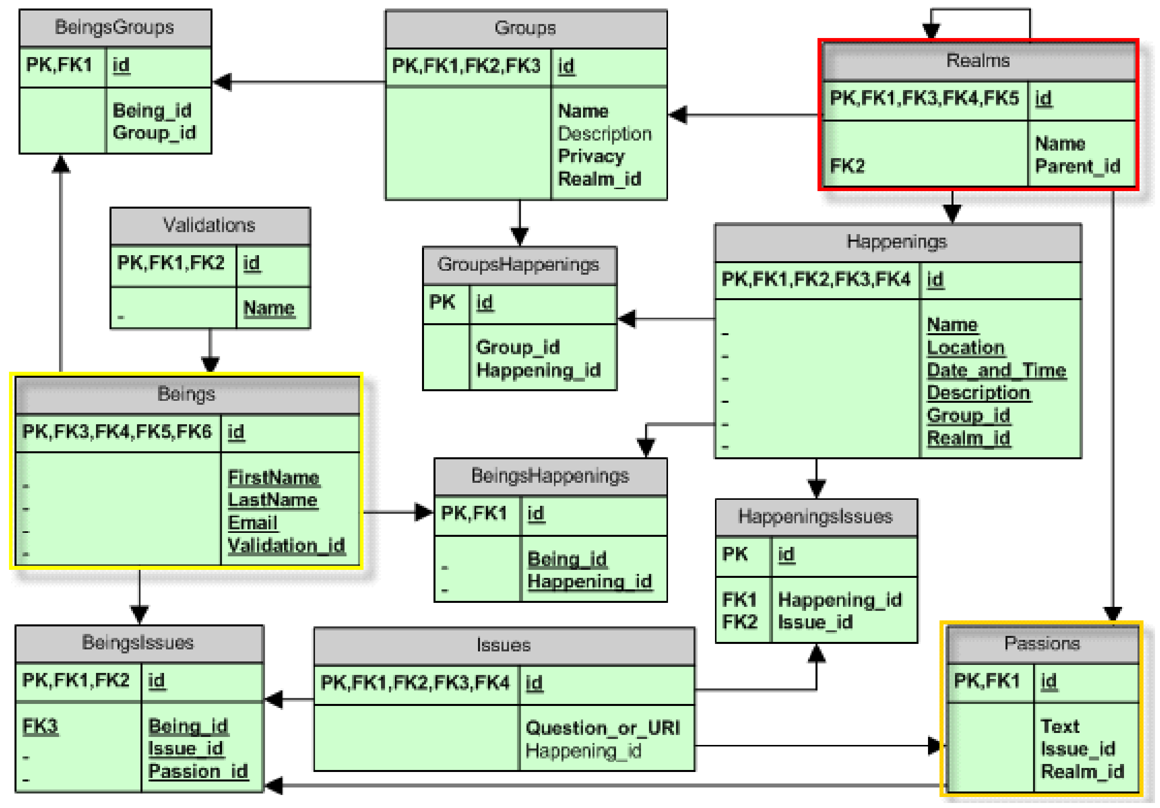
table
2. Active messaging - to alleviate delays associated with - ***Phase 3***
3. synchronous outbound emails, and
4. broadcasts to groups - ***Phase 2***
5. XSLT rendering of a demonstrational lense for viewing a sample profile and/or group of profiles **-** ***Phase 2***

Rails Screens

1. Profile registration - Index / Create / Update / Login (very basic login implementation in Phase 1)
2. Profile definition
    1. Realms - Index / Create / Update
    2. Issues - Index / Create / Update / Search
    3. Passions - Index / Create / Update / Search
    4. Post tags (existing or new) to user profiles
        1. Streamlined screen for "rapid creation and posting of a tag"
    5. Users can (be invited to) answer
        1. an individual question, or
        2. a group of questions
3. User group - Index / Create / Update
    1. Associate a user with (invite a user to) a group
    2. View list of users in a group
4. Profile viewing as XML data sets
    1. a single person's profile, or
    2. a group of users's profiles
    3. demonstration viewer to render the XML data in a user-friendly format

Overview Stories Data Model Crows Philosophy Geek Notes

The Data Model underlying "InterEthos"



Note: There is also a "Users" table, in addition to the above tables. It was installed with the Restful Authentication plugin. It is not used by the other tables, and only serves logins.

Overview Stories Data Model Crows Philosophy Geek Notes

Crows

This project not only included the building of a complex set of data-driven functionality, but a 100% "soul searching" discovery and definition of what that functionality was. I have never seen or used it before (no one has). So this was an excercise in invention, on top of data-driven web programming.

"Act as Tree" was a major plugin, which I customized with:

1. adding more layers of branching
2. adding a session "session[:uroot_id]" variable, which is set by an "...&uroot=#" url variable, and which is used to keep the branch of the "Realms" taxonomy you were last working on expanded. That way you do not come into the "index" Realms view, where all the branches are collapsed by default, and have to keep opening up the branch. You might be in process of adding a new branch having fifty or so sub-realms, and so having to keep opening the branch you are working on is not tolerable. I have implemented that session variable on the "change (triangle icons)" and "add (+ icons)" but ran out of time to do it on the "delete (- icons)".

Horizontal peach-colored toolbar with "page anchor links" that displays multiple times between the sections of the "About" page

This is an original application totally pulled from my "groping" imagination.

1. I had no pre-existing data models that I could copy from or build upon.
2. The final data model of this project is more complex than I should have taken on. Then again, I had to do it. I was possessed.
    1. It has eleven tables as compared to the seven in MetricsMine.
    2. In particular I wrestled many hours with the "Issues / Passions" relationship, migrating my code from a simple parent-child two-table model to a three-table HABTM model and finally back down to the simpler parent-child model over a period of weeks. I am still not sure that a Passion should not have the flexibility of being applied to more than one Issue in the broader concept of measuring and sharing the spirit of humans. I wrestled with this for many hours and days and weeks.

Data model diagram in "About" page

1. I spent many hours just staring at the data diagram, and at the navbar links of the web interface, trying to imagine the user experience and data flow.
2. I also spent much time "mastering" MS Visio to the point that I could keep generating updated "screen snaps" of the data model. This was Key to my being able to conceive of the project's final implementation.

At one point I totally destroyed all everything... (code, tables, data diagram), and renamed all the InterEthos table objects (and all related collateral), re-scaffolded everything, and merged the old code back into that new. This effort was done so that the "civilian" users of InterEthos might enjoy a better choice of words describing the various functional aspects of InterEthos, as well as that the envisioned "legions" of open-source code developers might survive **an "alphabetized" heirarchy of the ten (10) tables linked by the four HABTM relationships.** In Rails, the layout of HABTM data objects is so crucial for developers to internalize (especially given the total absence of any SQL statments). With the following new more vibrant names, and with the alphabetizaton, users and developers will hopefully more fully embrace and foster the structure of InterEthos.

Here is the alphabetical listing of the core tables:

1. **Beings**
2. **BeingsGroups**
3. **Groups**
4. **GroupsHappenings**
5. **Happenings**
6. **BeingsHappenings** *(actually reaches to side of nose-to-toes path, back to the "Beings" table.)*
7. **HappeningsIssues**
8. **Issues**
9. **BeingsIssues** *(this table cuts off the nose-to-toes path, allowing the cardinal relationship of Beings to Issues and Passions.)*
10. **Passions**

There are other tables as follows, but they do not serve in the cascading nose-to-toes dependent relationship of the above tables:

1. Users *(serves user logins, and is not to be confused with "Beings")*
2. Validations
3. Realms *(connect across all of the object tables to codify the ethos that happens in "Groups", "Happenings" and most importantly "Passions".)*

Of course we expect the number of table objects to change (increase) over time, but at least this core of ten names is alphabetical... and vibrant.

"Data_Add..." migrations via integration of "ar_fixtures" plugin commands in the "

Navigation - On top of a changing data model (described above), I implemented a lengthy drill-down menu through "Beings ===> Groups ===> Happenings ===> Issues ===> Passions". The user experience of this set of functionality is "hairy". There are so many branches to how the user might want to click and edit existing and create new records across that toolbar. You just have to click around for yourself. I did so until I couldn't do so any more!

Third party plugins, in order of implementation

1. Acts_as_tree
2. Acts_as_list
3. Ar_fixtures - "dump_to_file" and "load_from_file"
4. Action_mailer - *have not done anything functional with this yet.*
5. Restful_authentication

Overview Stories Data Model Crows Philosophy Geek Notes

The Philosphy of "InterEthos"

It must...

- ... be useable by everyone, and yet owned by noone
- ... cataolg the spirit of Individuals,
  including those who are not associated with Groups
- ... be totally "Spartan" in appearance, and extremely simple to use
- ... be elemental and monolithic, in its presentation of collective human spirit
- ... impose an "aura of majesty" on noone
- ... not rely on users commanding "fancy" technologies, but only the simplest and most ubiquitous
- ... be built on world-class technologies that
  enable security, speed, and richness of spiritual sharing
- ... be available for download in its entirety, including its full documentation
- ... be very easy for other people to replicate in its entirety to other servers
- ... have a method for integrating the harvested data to a shared "cloud" repository,
  should the implementor and beings-individually be so inclined

Overview Stories Data Model Crows Philosophy Geek Notes

Geek Notes

On migrations all of the test data will reload automatically, in all tables. If you run MIGRATION, then your down migration of
rake db:migrate version=0110
then that will preserve all login records in the "Users" table. Otherwise you will have to go throught the Restful Authentication process again, with email and all. Also, to save time you might want to leave the "Realms" table data with
rake db:migrate version=0210
as that takes time to unload and load.

Following are the notes kept while building the original implementation. Note that the application relied implicitly on "scaffolding", and changed the generated formats only slightly to include things like pull-down menus, redirects on "create" and "update" redirects, and necessary plugins.

Scaffolding
Note: the chronological and indentation of the following scaffold commands attempts to show dependencies such that the more indented tables depend on (ie. reference) the less indented tables

- ruby script/generate scaffold realm name:string parent:references
- ruby script/generate scaffold validation name:string
- ruby script/generate scaffold being first_name:string last_name:string email:string validation:references
- ruby script/generate scaffold group name:string description:text privacy:integer realm:references
    - ruby script/generate scaffold BeingsGroup group:references being:references
- ruby script/generate scaffold happening name:string location:text date_and_time:date description:text group:references realm:references
    - ruby script/generate scaffold GroupsHappening group:references happening:references
    - ruby script/generate scaffold issue question_or_uri:string happening:references
        - ruby script/generate scaffold HappeningsIssue happening:references issue:references
    - ruby script/generate scaffold passion name:string issue:references realm:references
        - ruby script/generate scaffold BeingsIssue being:references issue:references passion:references

Plugins

These plugins are listed in the order of implementation.

- ruby install git://github.com/rails/acts_as_tree.git
    - documented on / installed from: http://github.com/rails/acts_as_tree/tree/master
- ruby script/plugin install git://github.com/rails/acts_as_list.git --force
- script/plugin install http://topfunky.net/svn/plugins/ar_fixtures
    - ruby script/runner "User.dump_to_file"
    - ruby script/runner "User.load_from_file"
- ruby script/plugin install http://code.openrain.com/rails/action_mailer_tls

- Restful Authentication
  - http://github.com/technoweenie/restful-authentication/tree/master
- ruby script/destroy scaffold Way_Point
- http://github.com/swithin/interethos/tree/master
- rake routes
- ruby script/plugin install http://redhillonrails.rubyforge.org/svn/branches/stable-2.0/vendor/plugins/redhillonrails_core
- ruby script/plugin install http://redhillonrails.rubyforge.org/svn/branches/stable-2.0/vendor/plugins/foreign_key_migrations
- ruby script/plugin install http://redhillonrails.rubyforge.org/svn/branches/stable-2.0/vendor/plugins/foreign_key_associations
- to run these plugins
  - rake foreign_key_associations
  - rake foreign_key_migrations

Logout **Realms** About