Design and Implementation of The Nanenda (Nano-Agenda) PIM.

Edmund Ronald edmundronald@gmail.com

The soul of Nanenda.

This note describes my design process and implementation steps for Nanenda (Nano-Agenda), a simplified notetaking package inspired by Agenda. As Nanenda can be viewed as a reimplementation of some functionality from Agenda, we give a brief analysis of Agenda, the data structures which I chose to implement from the start, and the process by which I simulated the use of Nanenda before writing the code.

Implementation Plan

(x) marks done, o(in progress), (.) planned

Item Data structure

The initial item datastructure will be <Annotation - Tags - URI>. This seems to cover 90% of what I personally use for research and everyday notes. Planning steps eg TODO can be added as tags and cause a trigger out to a Kanban, which seems a much better planning display anyway than Agenda ever was. I am not putting calendaring in the initial software.

Core Program

Ideas ==> Use pattern ==> Spreadsheet Simulation ==> Code+Spreasheet ==> Standalone App.

(x)Ideas - see below - from Lotus Agenda.
(x)Use pattern
(x)Spreadsheet simulation
(o)Code+Spreadsheet
(.)Standalone

Filters

Dumb filters ==> known-data pickers ===> smart filters by pattern matching ==> Word2Vec filters.

Basic decisions

Nanenda is intended to be a single-person research and notes to self tool, with some modest task planning capabilities. It is centered around notes-to-self and user interaction is involved in annotating and tagging entered data. Thus, Nanenda is intended for research notes and long term information retention.

- *My fundamental decision was to minimise functionality in the core of Nanenda, while providing hooks for add-ons.* Keep the software tiny, open, and then extend with nifty external accessories to add functionality - eg. dictate voice notes from iPhones converted to text.

- My second basic decision was to use a data format that would be human-readable, as I don't want anybody including myself to be stuck in the future with notes that are locked away. The data should easily be exportable to CSV and read into a spreadsheet, providing good future-proofing.

- In fact, I think that just a core dataformat definition by itself and the understanding that the internals should provide hooks for external processing at every stage is almost enough to guide the creation of Nanenda, a reimplementation of Agenda. The core of Nanenda ends up tiny, much of the functionality is in peripheral software, and computer litterate users can write their own software to add to and view the database. It should even be possible to swap out the core itself, just as one can swap the engine of a car.

- The "smart" functions of Nanenda should be viewed as add-ons, not core. For instance if the right hooks are supplied to be triggered by the input module, then smart parsing should be viewed as an input plugin rather than a core function, as can mundane tasks such as timestamping every item entered. External software should of course provide all the bridges from datasources eg. email or web to Nanenda.

Historical Background.

Lotus Agenda (1992) was a groundbreaking complex and powerful Personal Information Management (PIM) package that was never transitioned to the Windows era and ultimately disappeared from use.

Although it was long cited as an extremely useful tool by its devotees. Agenda may have suffered from its mouseless dense free-form terminal-screen presentation which presupposed a learning curve and may have been appropriate only for executives and knowledge workers; as such Agenda was soon replaced in the Lotus lineup by the less powerful but graphically refined Lotus Organiser package.

The PIM category has since seen the appearance ***and disappearance*** of a multitude of well accepted, some even say much loved, mainstream PIM and outliner packages such as Lotus Organizer and MORE, as well as the emergence of niche tools for writers and researchers; the reason why many of these third party packages tend to a limited life-cycle is not obvious, but may be linked to the proliferation of external data data silos eg. enterprise CRM systems, and/or bundled competitors with tangential outlining or data management functionality such as Powerpoint and Outlook, and of course the integrated software supplied by Google and Apple for their phone ecosystems.

A study of the reasons why this category of software has seen such a flurry of births and deaths might make an interesting foray into computer archeology, but we think it is faster just to implement Nanenda and continue our lives.

Present Day Context.

Modern day dedicated PIM and notes software mostly presents as SAAS, with some niche exceptions such as Zoot which may provide synchronisation between computers or external backup but do not rely on the cloud for primary data storage.

SAAS provides multiple advantages for the user, eg install-free onboarding, but it also creates unease when essential digital property such as PhD research notes or book drafts move away from the users devices to a system that might be accessed by third parties or even shut down for economic reasons.

Analysis: Function Modes of Agenda.

Due to its DOS and pre-web and pre-email background, Agenda was basically a standalone piece of software. Direct user interaction was the main intended mode of data acquisition, although the program could import data via a screen-clipping facility, and ingesting files via converters. Data display was primarily intended to be effected through the screen, and printout.

In use, for our purposes, Agenda can be considered as functioning either
- in data entry and smart parsing mode as items are being added or edited
- in viewing mode,when perusing data that is present
- in organisation/housekeeping mode when category or view structure is being edited.

Three types of interacting data structures facilitate the Agenda experience:
- Items - these are the fundamental molecules of information which are stored.
- Categories - these are used to impose structure over collections of items.
- Views, for selecting and presenting collections of items for display.

In summary, Agenda's editors/displays are for items, categories, views.

Decisions on the scope of NANENDA.

We believe that Nanenda should at least initially embrace the obsolete model of local data with remote sync facilities rather than SAAS, and implement core functionality suitable for “slow” data use such as notes to self and research notes. The research could involve design - this text was researched using a simulation of Nanenda - or the material for a novel, or even the creation of cooking recipes.

Thus Nanenda can be viewed as an aid to both short term memory for the span of a work session, and long term memory as measured in months or years, rather than a tool for optimising email or business response. Nanenda is basically an electronic version of a bound notebook or ring-binder rather than a stack of throwaway yellow Postit notes.

We are not initially prioritising the sophisticated calendaring features which were provided in Agenda, although we intend to provide some rudimentary task management, ie KanBan board as an accessory..