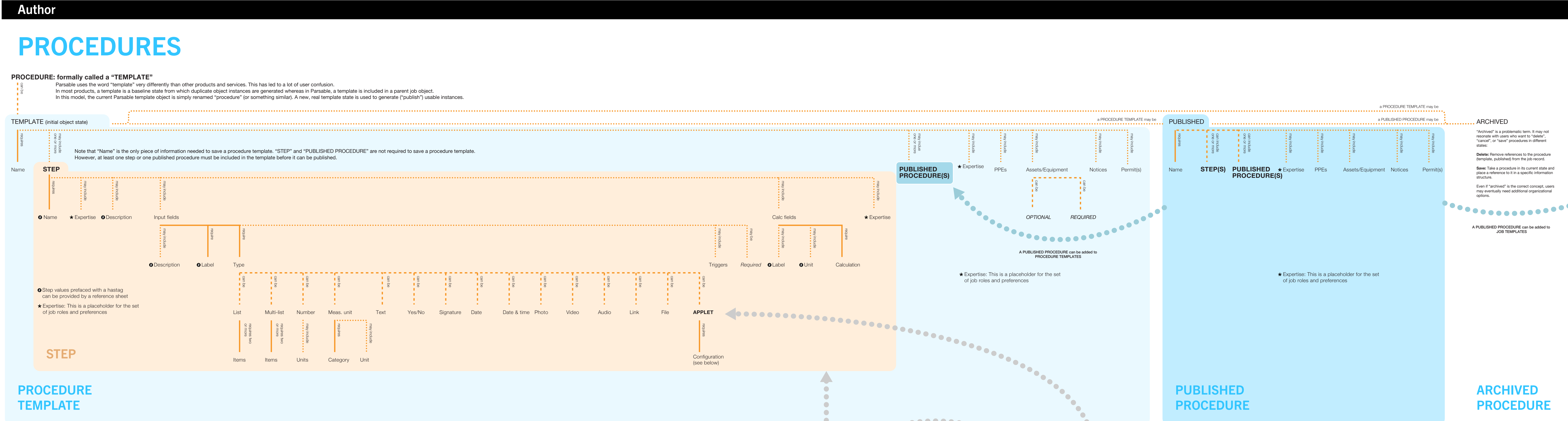


Parsable User Object Hierarchy



THE PROBLEMS

No clear representation of hierarchy: The Parsable object model is strictly hierarchical. However, the web interface presents Parsable objects (reference sheets, applets, templates, jobs, and workflows) as peers. Authoring options are non-intuitive (adding templates to a job rather than creating a job from a template; restricting roles to certain people rather than assigning roles to people).

Non-standard terminology: For example, in most products, a "template" is typically an element that generates unique instances of itself. In Parsable, a template is something added to a job—an object which itself does not have a true template state. A DUPLICATE function is used instead to create unique job instances. "Role" means different things in different processes.

Unclear object states: Parsable objects do not have appear to have states that logically correspond to concepts such as "draft" or "underway," "approved," "cancelled," or, "saved."

THE PROPOSAL

- Design an object model which is easier to explain to users yet is based on the current product, namely:
- In Parsable, a **PROCEDURE** contains the **STEPS** (and other PROCEDURES) needed for a **JOB**.
 - A **JOB** contains one **PROCEDURE** as well as a schedule for the job.
 - A **WORKFLOW** contains a set of **JOB**s as well as a schedule for the overall workflow.
 - Users author **TEMPLATES** which can generate active **PROCEDURES**, **JOB**s, and **WORKFLOWS**.

In this model, every object has a template state as well as an active (e.g. "published", "open", etc.) state. Incorporating an active object into another object is akin to placing an image in a presentation—the original remains intact, and a copy is added to the destination. This model also supports linking, which would allow an author to edit an active object and have those edits appear everywhere that object is referenced.

KEY

