How Do You Design the Future? 2012–2013 Lecture Series
CMU School of Design / HCI
3 October 2012, Pittsburgh

A Systems Perspective on Design Practice

Hugh Dubberly
Dubberly Design Office

Main Points

The future of design involves **systems**
Designing systems involves **models**

What models are necessary + sufficient?

Concept maps can describe many systems
Conceptual models aid software design

We are in the midst of a fundamental shift in how we view the world.

from

Mechanical

to

Biological

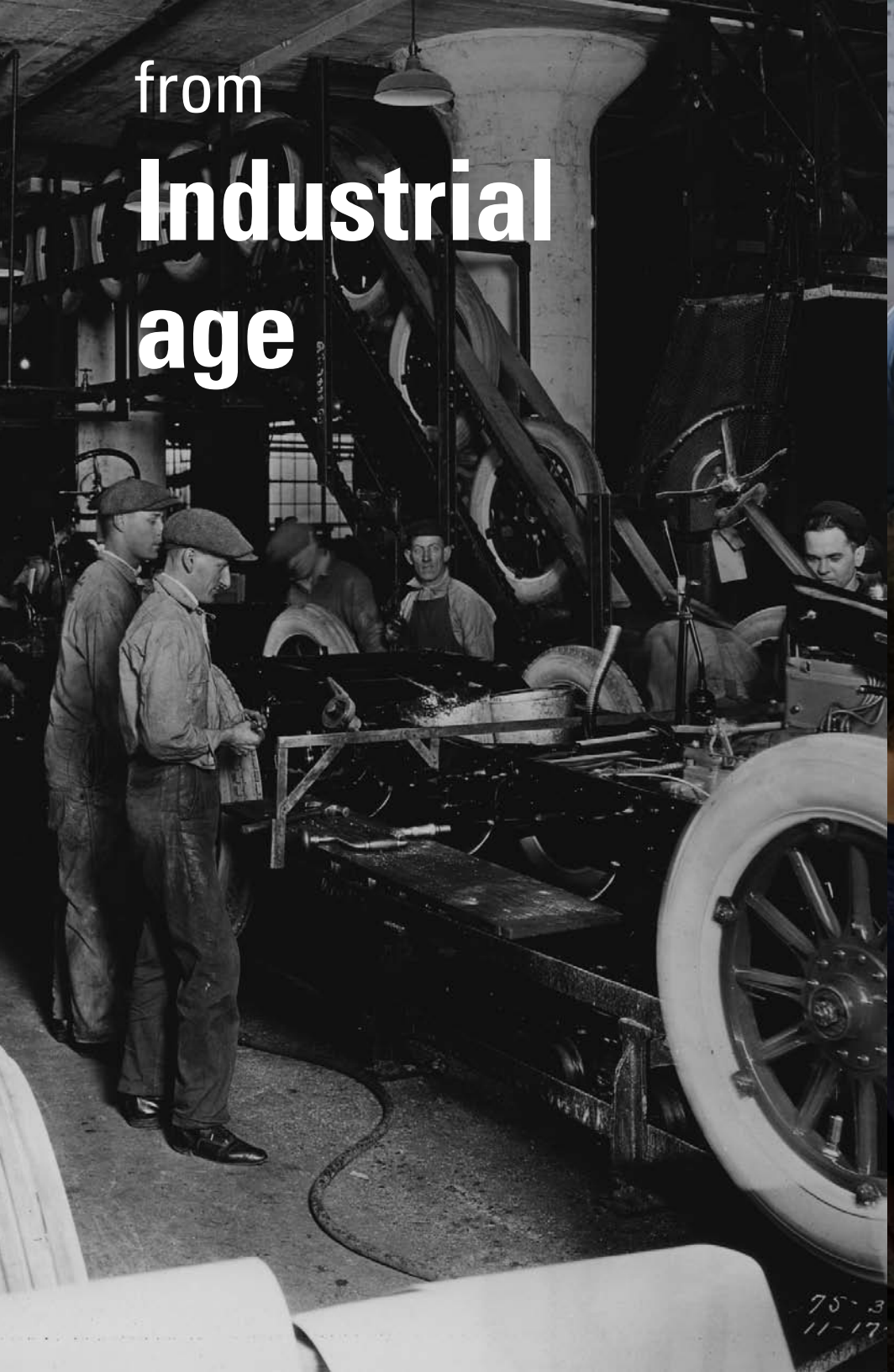
from

Newton

to

Darwin

from
Industrial age



to
Information age

The shift in world view coincides with a shift in our view of products.

"... commercial products are best treated as though they were services.

It's not what you sell a customer, it's what you do for them.

It's not what something is, it's what it's connected to, what it does.

Flows become more important than resources. Behavior counts."

— Kevin Kelley, *Out of Control*

Thinking in terms of whole systems means

- Building **relationships** between products
  e.g. roadmaps, product lines, platforms, APIs
- **Continuous change** + dynamic development
  e.g. stocks, flows, lags, oscilation
- Enabling **feedback**
  e.g. goal-action-measure-compare loops
- Adopting **metaphors from nature**
  e.g. ecology, evolution, emergence

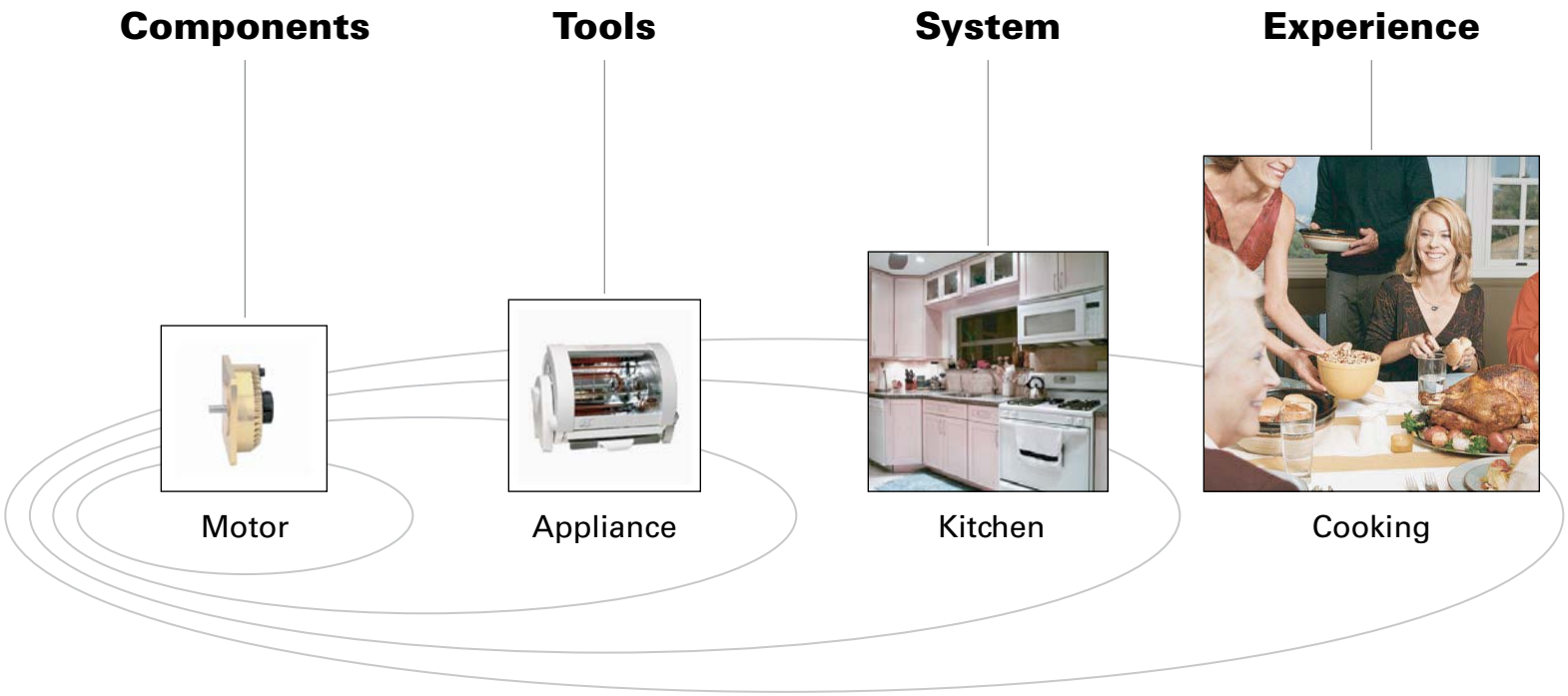
Systems affect many dimensions of design.

- Creating and managing (networked) services
- Connecting products + services
- Integrating across products
- Building a seamless brand experience
- Communicating with consistency
- Creating a sustainable business (green design)

Hardware products are increasingly tied to:

- embedded software
- the internet and web-based applications
- human services
- the organizations which develop and deliver the products and services
- communities for which they provide infrastructure
- the ecologies in which they cooperate and compete

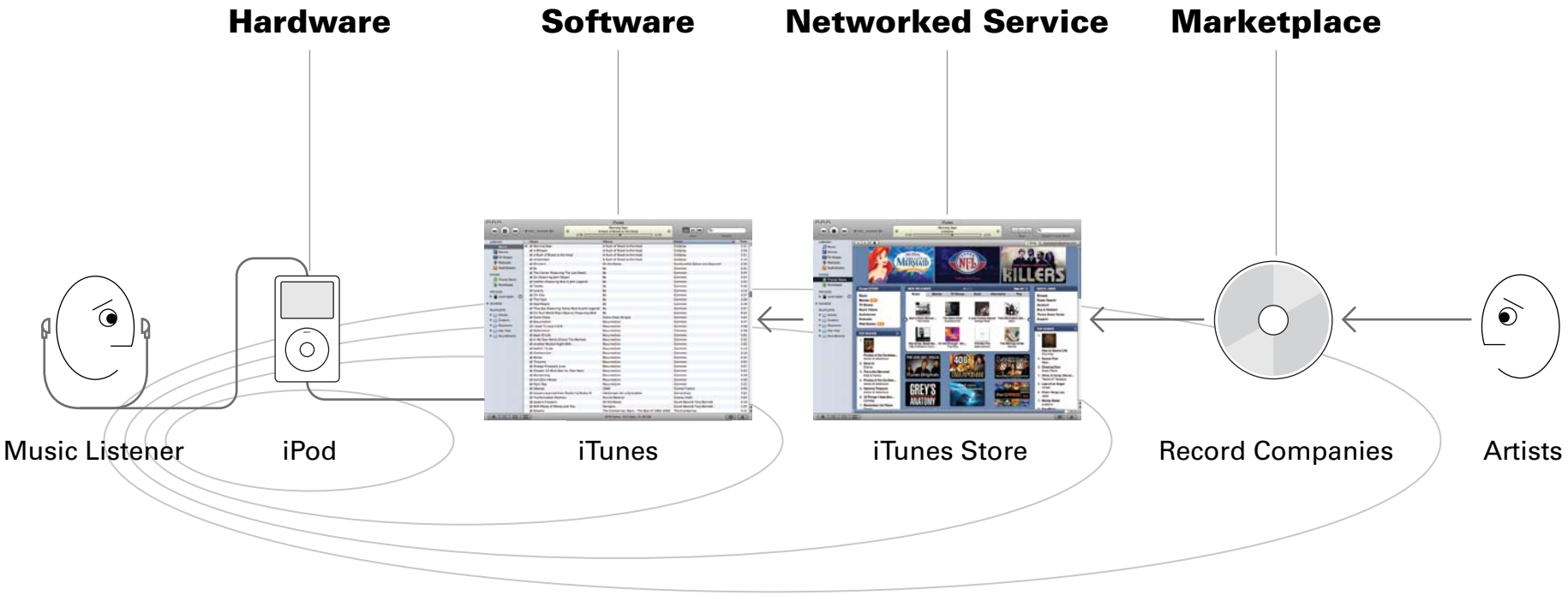
Value comes from interacting with larger systems—enabling an ecology.



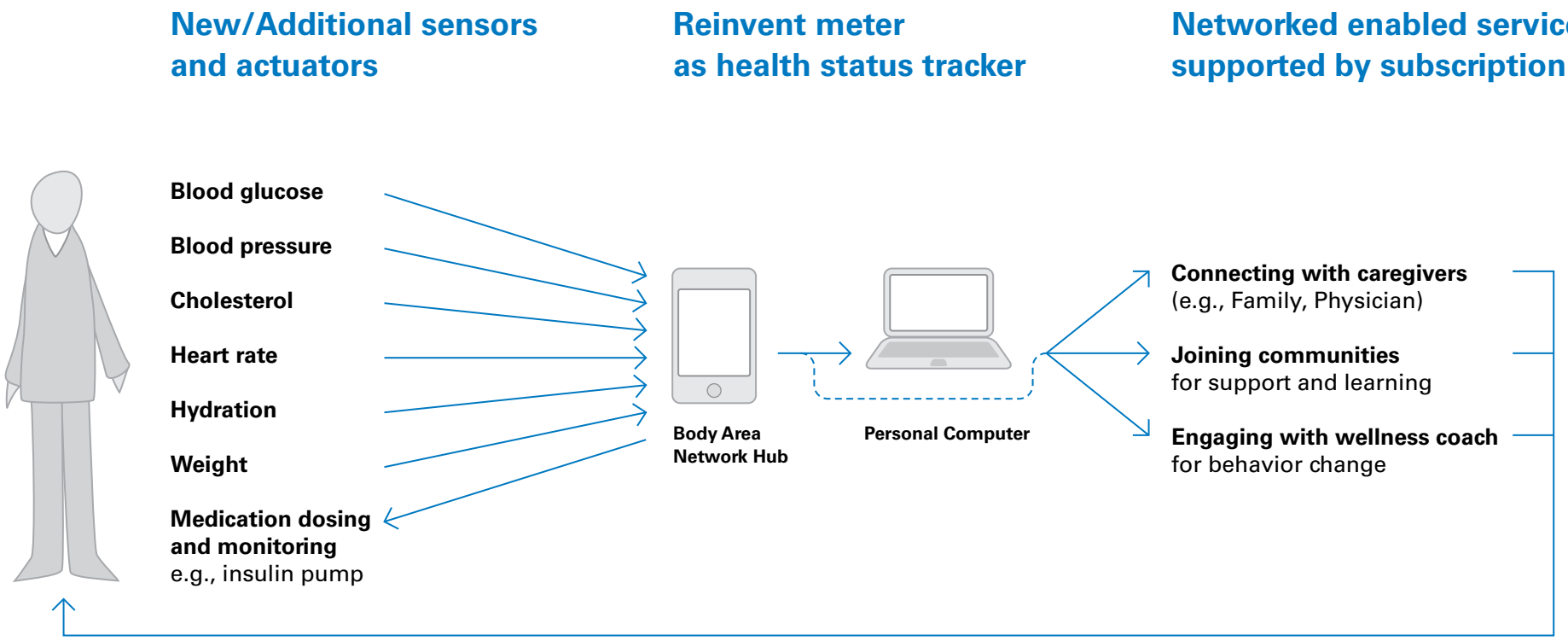
— John Rheinfrank & Fred Murrell

iPod is an integrated system.

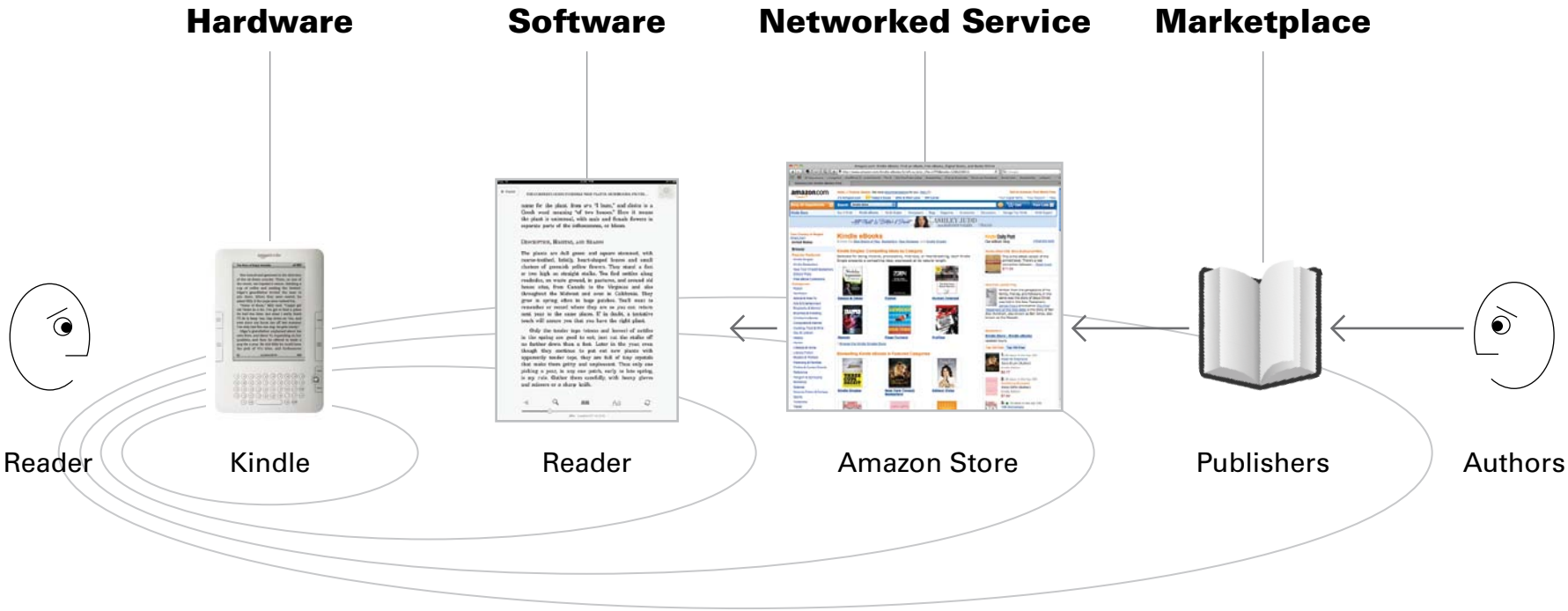
DRAM > mp3 player > music sharing service > my music



In just a few years, iPhone and other smartphones will become hubs of body-area networks.



Amazon's Kindle-Reader-WisperNet-Store system is another networked-services ecology.



"I think of [the Kindle] as a service.
Part of [it] is of course the hardware,
but really, it's the software, the content,
it's the seamless integration of those things."
— Jeff Bezos

The shift
in the nature of products
requires a shift
in the way we design.

	From ... escaping the past	To ... inventing the future
	Manufacturing Age	**Age of Biology**
Focus	**Objects/Things**	**Systems/Behaviors**
Values	**Seek simplicity**	**Embrace complexity**
Designer's role	**Expert/Deciding**	**Collaborator/Facilitating**
Construction	**Direct**	**Mediated**
Stopping condition	**Almost perfect**	**Good enough for now**
Result	**More deterministic**	**Less predictable**
End state	**Completed**	**Adapting continuously**

Design education focuses on the form of objects; much of practice does likewise.

How are we making it?
Form/Grammar
Syntactic

Object
Component

Form is governed by meaning and structure, though they are also affected by form.

What are we making?
Meaning/Definition
Semantic

How are we making it?
Form/Grammar
Syntactic

Object
Component

Meaning + structure are governed by context; context is also affected by meaning + structure.

Why are we making this?
Context/Need
Pragmatic

What are we making?
Meaning/Definition
Semantic

How are we making it?
Form/Grammar
Syntactic

Object
Component

Objects are often embedded in systems.

Why are we making this?
Context/Need
Pragmatic

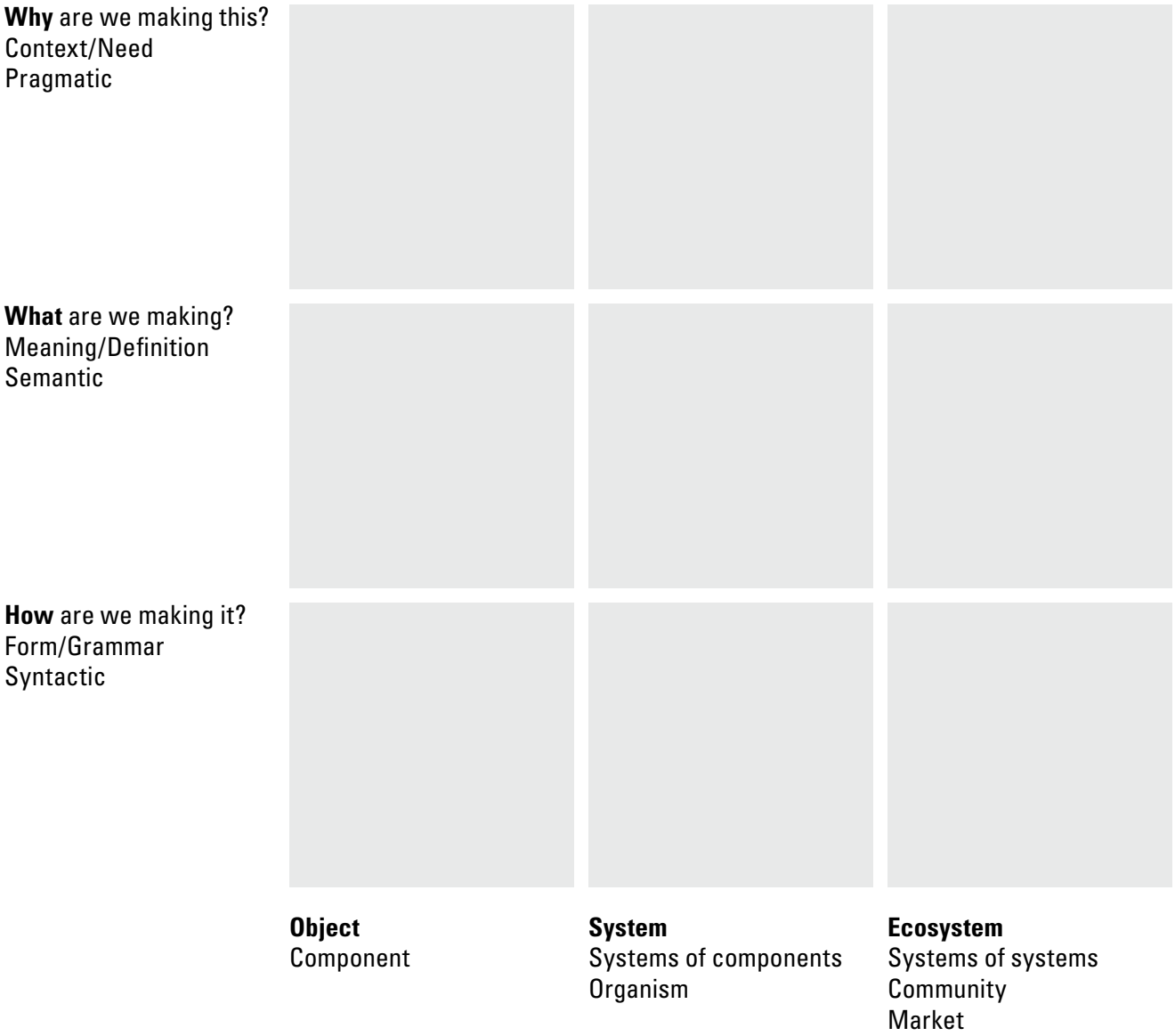
What are we making?
Meaning/Definition
Semantic

How are we making it?
Form/Grammar
Syntactic

Object
Component

System
Systems of components
Organism

Systems are often embedded in ecologies—communities of systems.

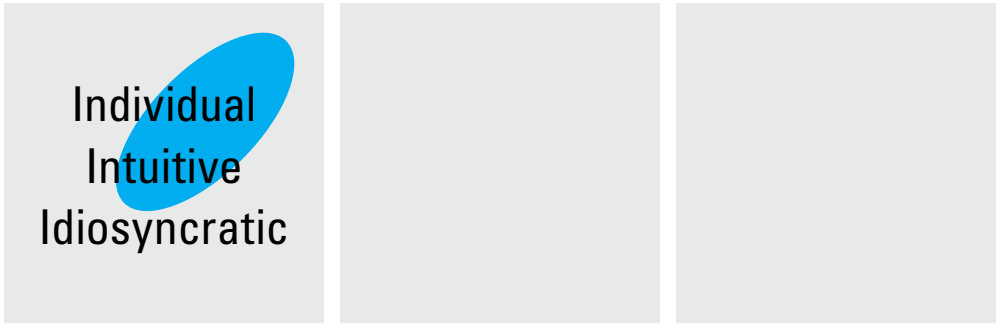


Practice focused on the form of objects can be direct and unmediated.

Why are we making this?
Context/Need
Pragmatic

What are we making?
Meaning/Definition
Semantic

How are we making it?
Form/Grammar
Syntactic



Object
Component

System
Systems of components
Organism

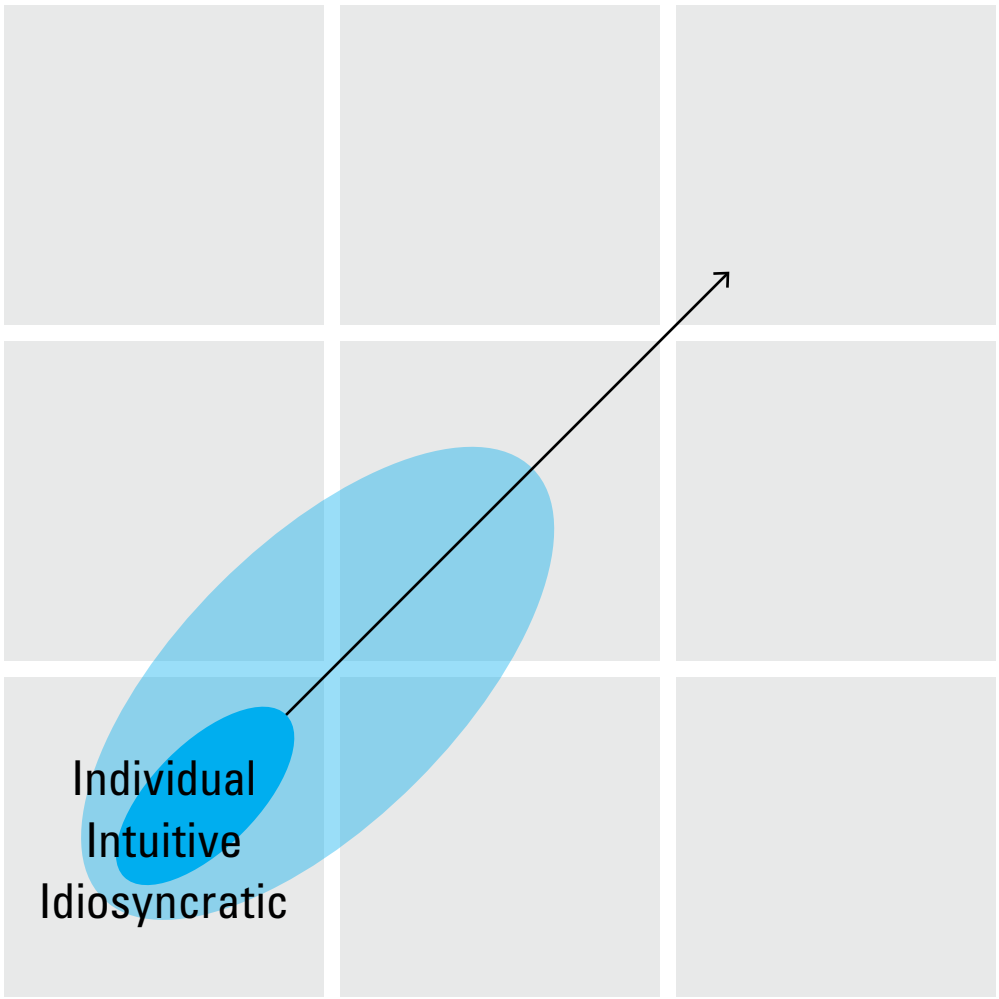
Ecosystem
Systems of systems
Community
Market

As practice expands, it becomes more complex.

Why are we making this?
Context/Need
Pragmatic

What are we making?
Meaning/Definition
Semantic

How are we making it?
Form/Grammar
Syntactic

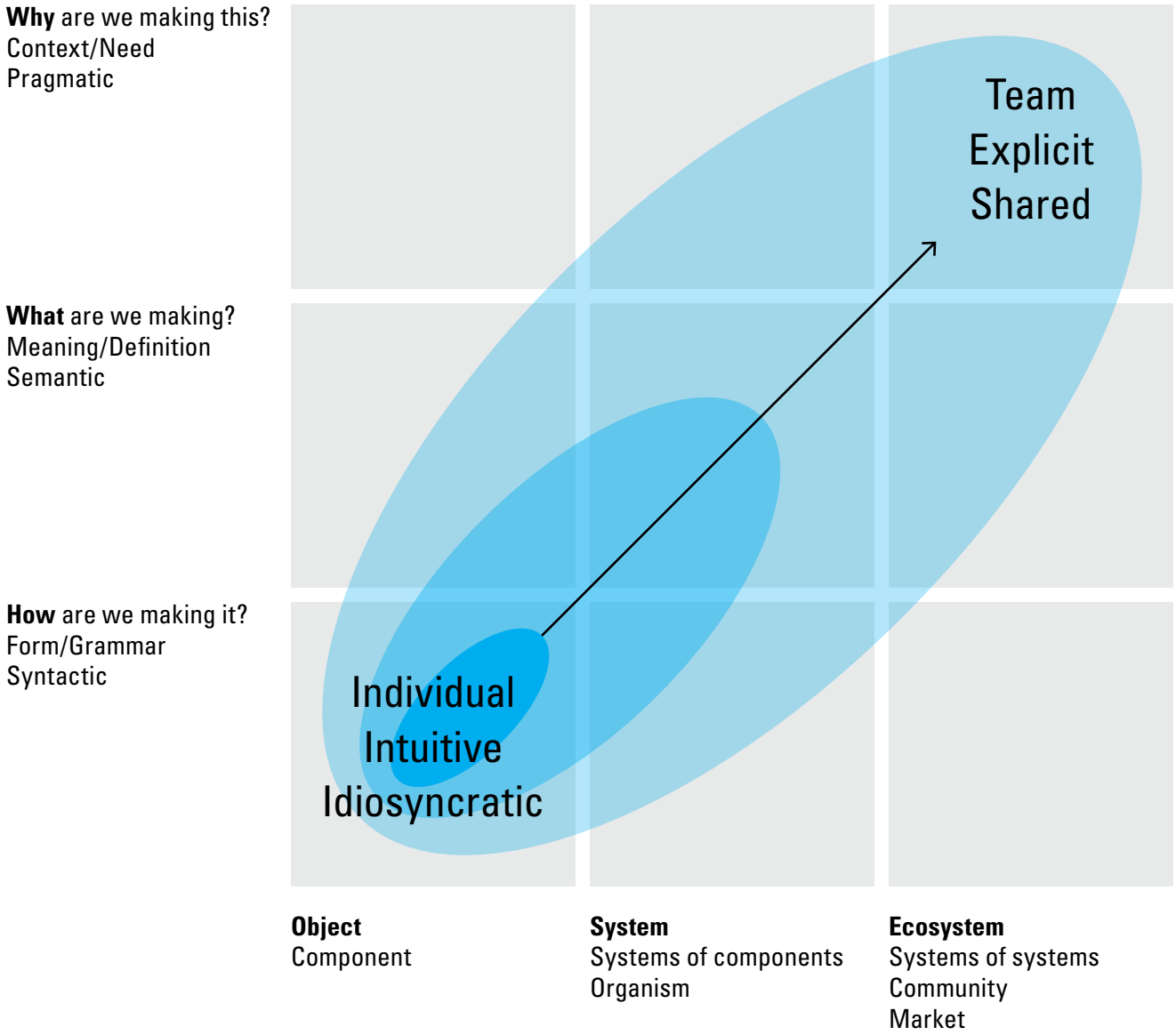


Object
Component

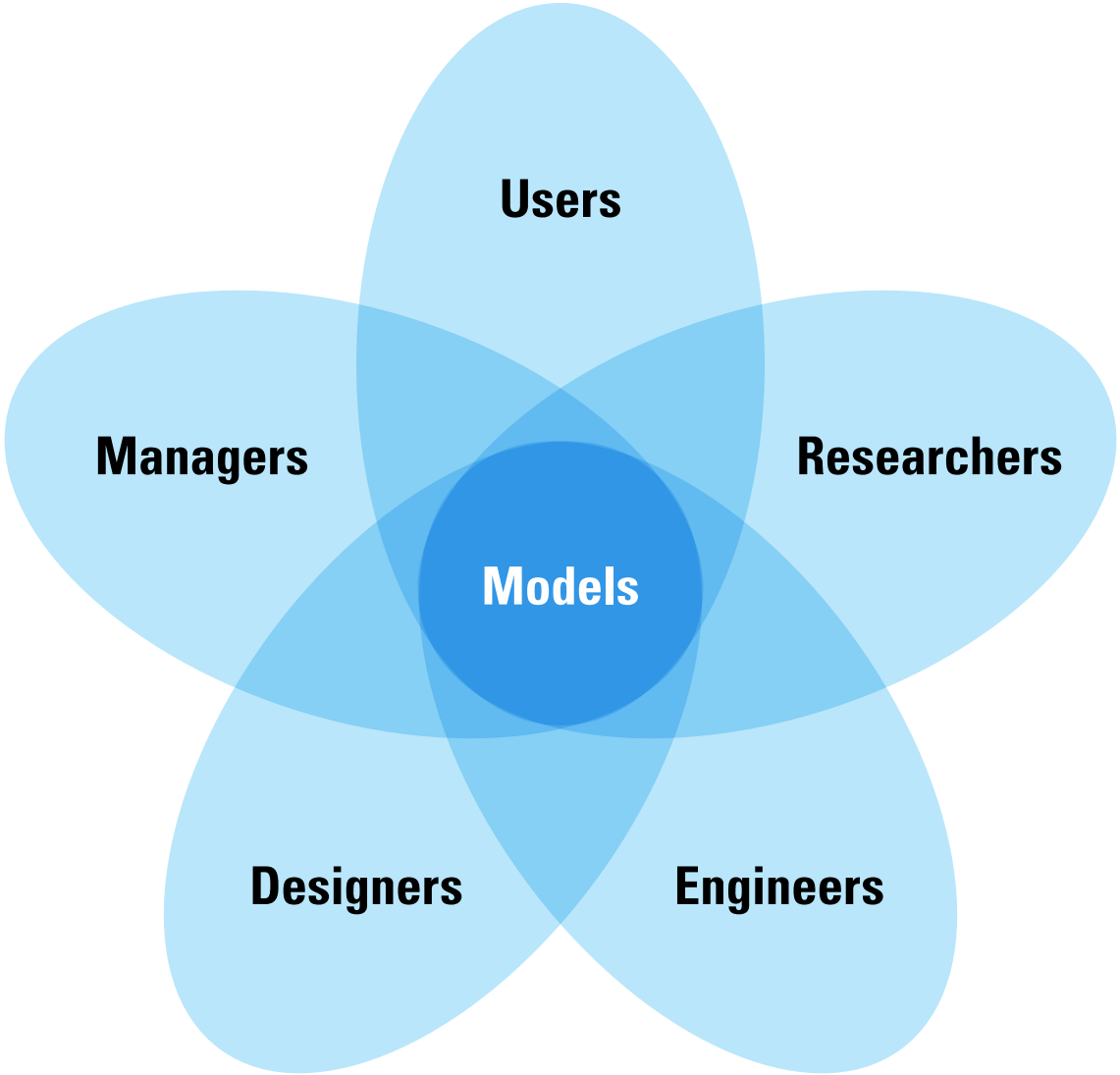
System
Systems of components
Organism

Ecosystem
Systems of systems
Community
Market

When practice also concerns context + ecologies, project teams require many disciplines.



For team members to understand each other, they need shared representations—boundary objects.



"Most scientific work is conducted by extremely diverse groups of actors . . . Simply put, scientific work is heterogeneous. At the same time, science requires cooperation*— to create common understandings, to ensure reliability across domains and to gather information which retains its integrity across time, space, and local contingencies."*

— Susan Leigh Star and James R. Griesemer,
"Institutional Ecology, 'Translations' and Boundary Objects"

". . . boundary objects are produced when sponsors, theorists and amateurs collaborate to produce representations of nature. Among these objects are specimens, field notes, museums and maps of particular territories. Their boundary nature is reflected by the fact that they are simultaneously concrete and abstract, specific and general, conventionalized and customized."

— Susan Leigh Star and James R. Griesemer,
"Institutional Ecology, 'Translations' and Boundary Objects"

What models are necessary + sufficient to create a new service or business?

Models

- Business model
- Organizational structure
- Product development process
- Manufacturing process
- Release process
- Marketing plan
- Distribution process

What models are necessary + sufficient to design a new software application?

Models

- Terrain map
- User opportunity/need model
- Primary user tasks
- Solution space
- Competitive space/positioning
- Service/system model
- Conceptual model
- Data model
- Information architecture
- Application architecture
- Network configuration

In order to design a software application, I need to understand the data model.

At their core,
all applications have a main data type,
and tools for selecting and changing data.

If you understand the main data type
and the tools for selecting and changing data,
then you understand the application—
or at least much of what it can do.

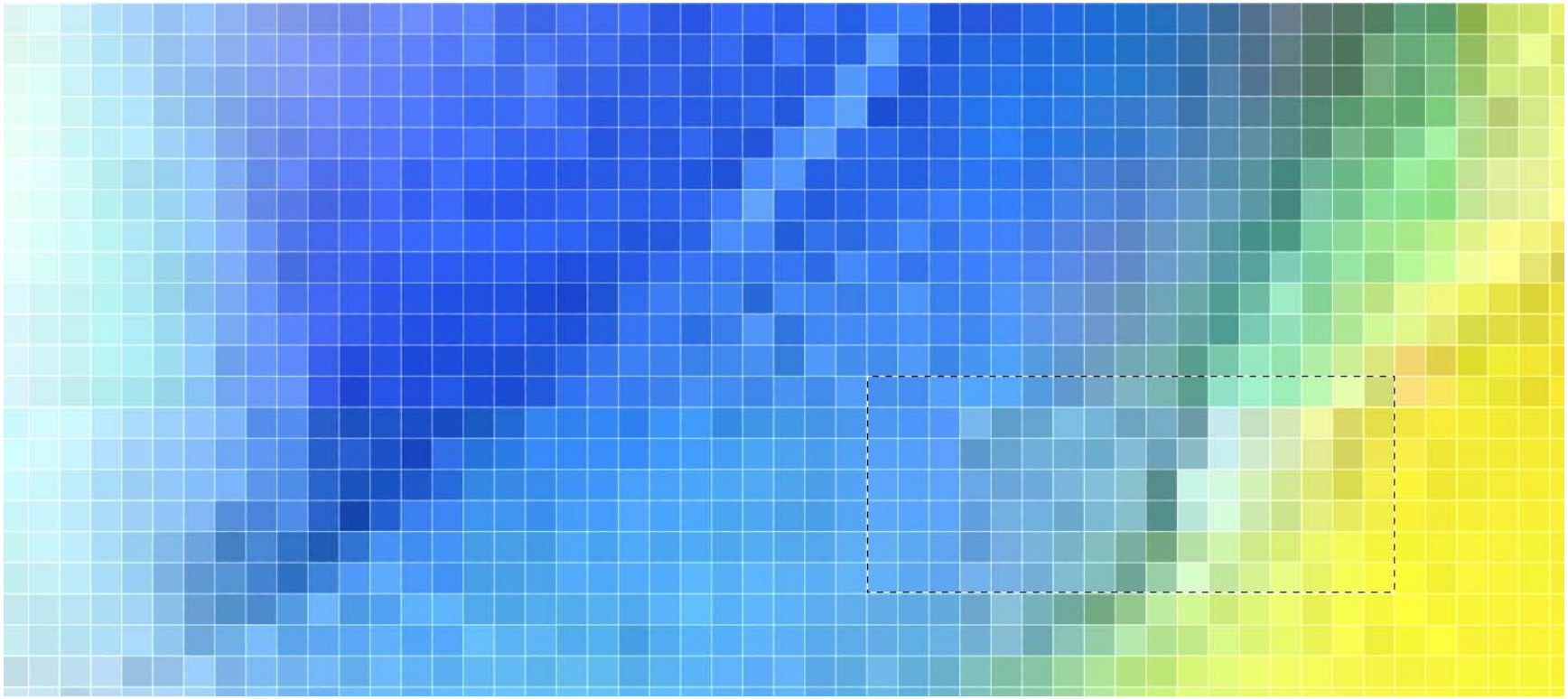
What defines a spreadsheet?

Spreadsheets are grids of cells,
containing text, numbers, and functions
that operate on a range of cells
This is what makes VisiCalc similar to Excel.

	A	B	C	D
1		**FY 10**	**FY 11**	
2	Jan	1	2	
3	Feb	3	4	
4	March	5	6	
5	April	7	8	
6				
7	**Totals**	16	=SUM(C2:C5)	
8				
9				
10				

What defines an image editor?

Images are grids of cells,
containing numbers (that represent colors).
You edit an image by selecting a range of cells
and applying a transform function.
This is what makes MacPaint similar to Photoshop.



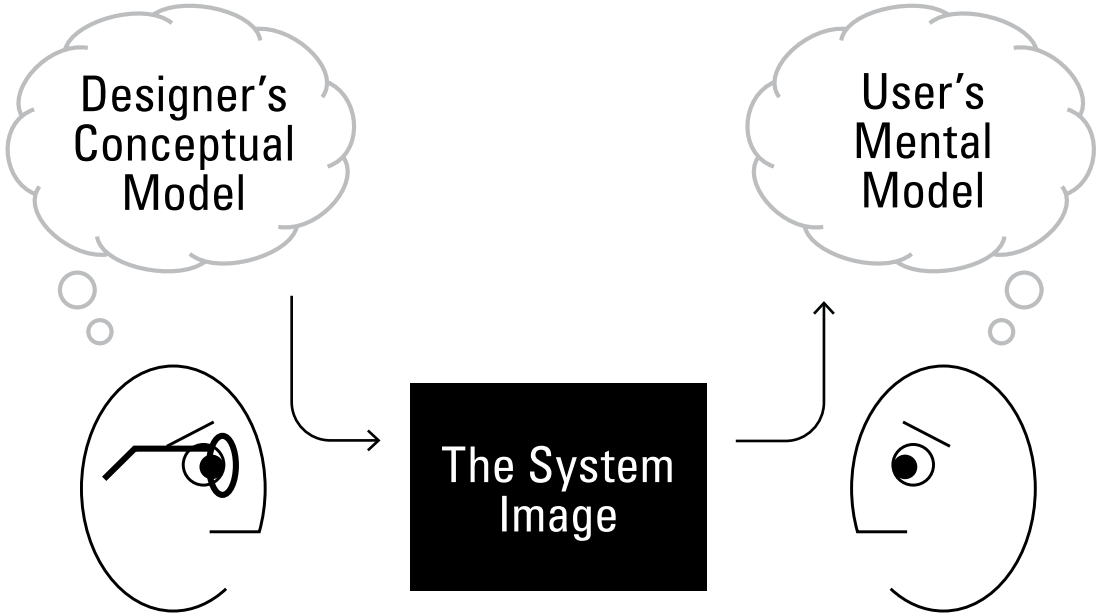
The data model is problematic for design.

- Difficult to teach + learn
- More about how than what or why
- Focuses on technology
- Doesn't focus on users

Don Norman talks about a similar idea:
The system image

"For people to use a product successfully, they must have the same mental model *(the user's model)* as that of the designer *(the designer's model). But the designer only talks to the user via the product itself, so the entire communication must take place through the 'system image': the information conveyed by the physical product itself."*

— Don Norman, *The Design of Everyday Things*, 1988



The system image is a nice idea, but it doesn't help us design.

But what if we made diagrams of the mental models we have or models we want users to have?

We have a tool for that—
concept mapping.

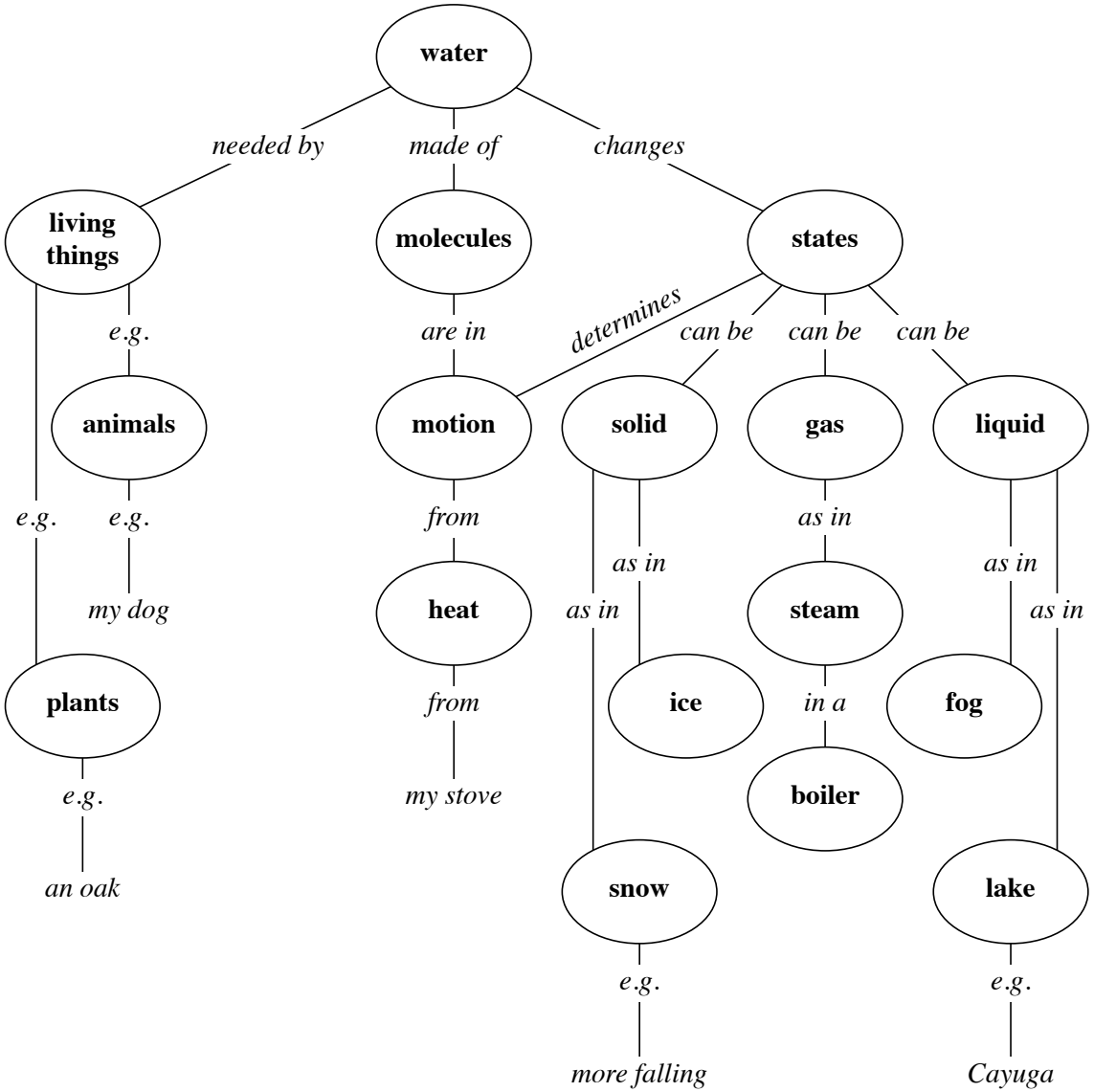
"Concept maps … represent meaningful relationships between concepts in the form of propositions. Propositions are two or more concept labels linked by words in a semantic unit."

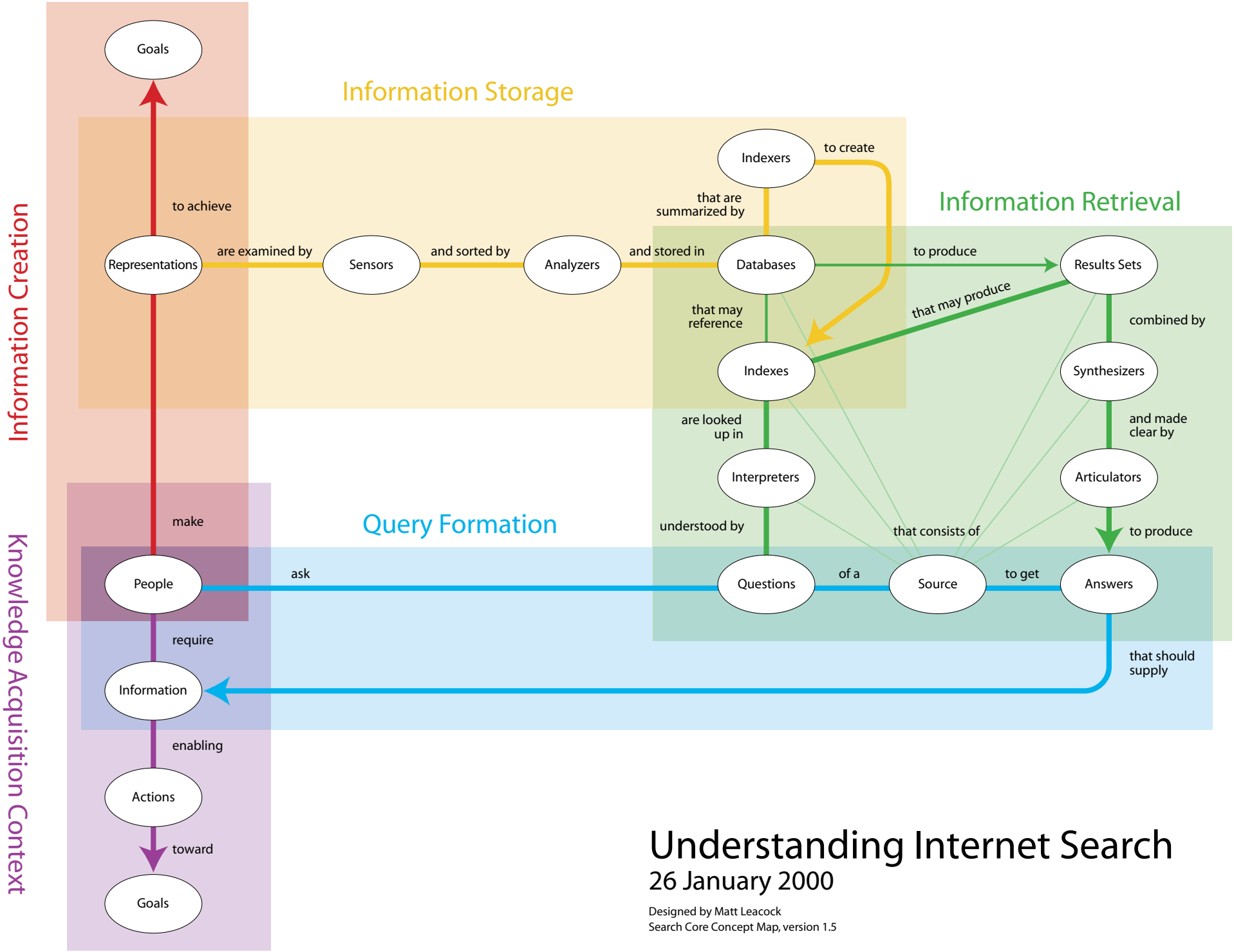
— Joseph D. Novak + D. Bob Gowin, *Learning How to Learn*

A proposition is a sentence.

Subject – verb – object

Node – link – node

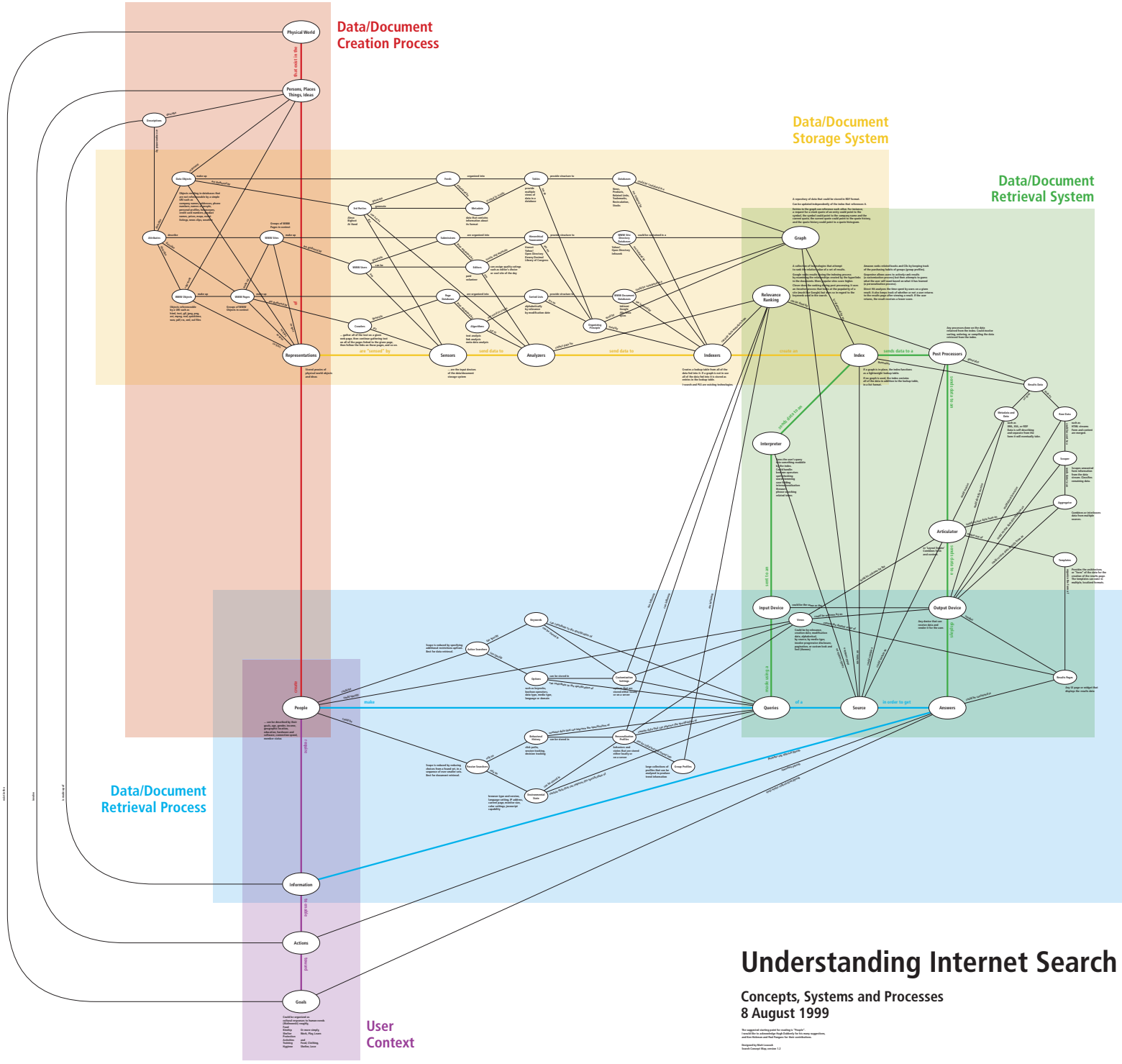




Understanding Internet Search

26 January 2000

Designed by Matt Leacock
Search Core Concept Map, version 1.5

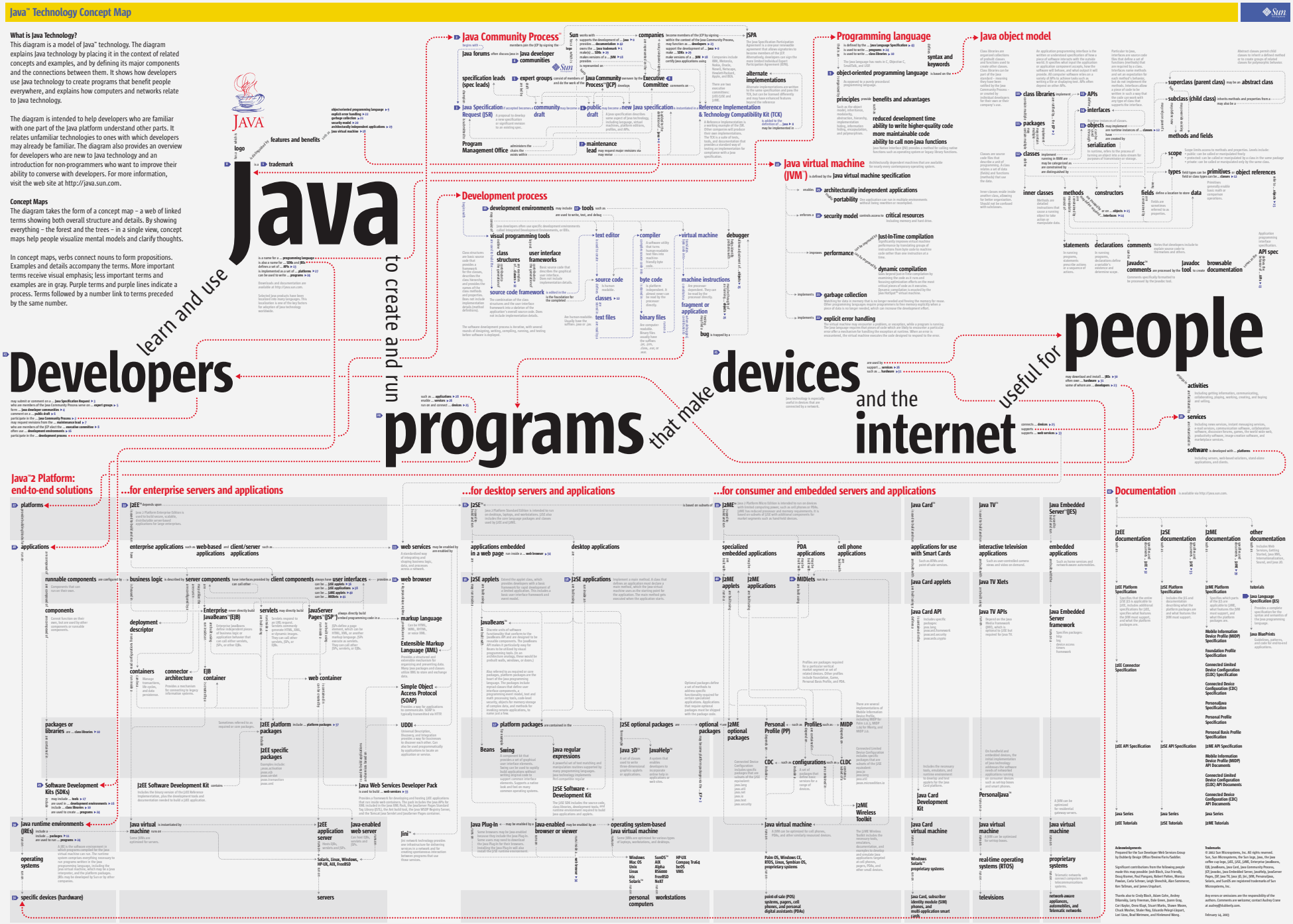


Understanding Internet Search

Concepts, Systems and Processes
8 August 1999

The suggested starting point for reading is "People".
I would like to acknowledge Hugh Dubberly for his many suggestions, and Eric Hickman and Paul Rangan for their contributions.

Designed by Mark Leonard.
Search Concept Map, version 1.2


Java™ Technology Concept Map
What is Java Technology?
This diagram is a model of Java™ technology. The diagram explains Java technology by placing it in the context of related concepts and examples, and by defining its major components and the connections between them. It shows how developers use Java technology to create programs that benefit people everywhere, and explains how computers and networks relate to Java technology.
The diagram is intended to help developers who are familiar with one part of the Java platform understand other parts. It relates unfamiliar technologies to ones with which developers may already be familiar. The diagram also provides an overview for developers who are new to Java technology and an introduction for non-programmers who want to improve their ability to converse with developers. For more information, visit the web site at http://java.sun.com.
Concept Maps
The diagram takes the form of a concept map – a web of linked terms showing both overall structure and details. By showing everything – the forest and the trees – in a single view, concept maps help people visualize mental models and clarify thoughts.
In concept maps, verbs connect nouns to form propositions. Examples and details accompany the terms. More important terms receive visual emphasis; less important terms and examples are in gray. Purple terms and purple lines indicate a process. Terms followed by a number link to terms preceded by the same number.
Java
Developers
learn and use
to create and run
programs
that make
devices
and the
internet
useful for
people
Java Community Process℠
Java object model
Programming language
Java virtual machine (JVM™)
Development process
Java™2 Platform: end-to-end solutions
...for enterprise servers and applications
...for desktop servers and applications
...for consumer and embedded servers and applications
Documentation

Concept mapping is a useful skill, especially for modeling systems, and it's easy to learn.

But how does concept mapping help with interaction design?

Interaction design consists of

- Concepts
  e.g., all concepts that the application's user interface exposes to users

- Taskflow
  e.g., the sequence of operations that users execute to accomplish tasks

- Presentation
  e.g., the controls, displays, etc. that comprise its user interface

— Jeff Johnson + Austin Henderson,
Conceptual Models: Core to Good Design, 2012

It's important to note
that the conceptual model is independent
of task flow and presentation.

As you work out task flow,
you may find problems in the conceptual model,
and you may need to modify it.

"A conceptual model is
a high-level description of an application.
It enumerates all concepts in the application
that users can encounter,
describes how those concepts relate to each other,
and how those concepts fit into tasks
that users perform with the application."

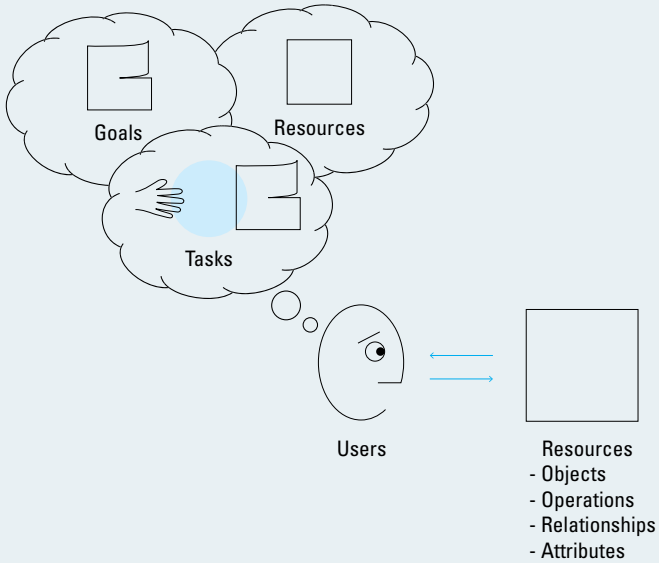
— Jeff Johnson + Austin Henderson,
Conceptual Models: Core to Good Design, 2012

A conceptual model includes,
all the data "objects,"
which a user may encounter,
and the "operations, attributes, and relationships,"
which a user may perform on the data objects.

Users focused on a task have in mind models of goals, tasks, and resources.

Contexts: Recent, current, anticipated

- Who (actors)
- What (resources)
- Where (location)
- When (time)
- Why (goals)
- How (activities)



Resources
- Objects
- Operations
- Relationships
- Attributes

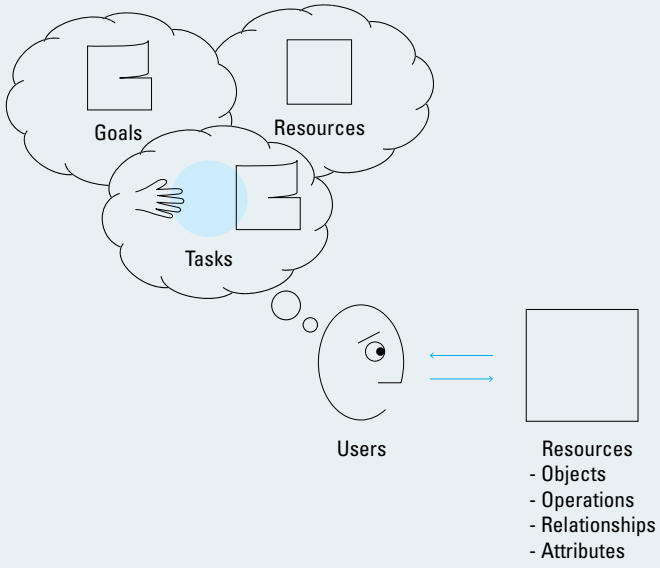
Task Domains

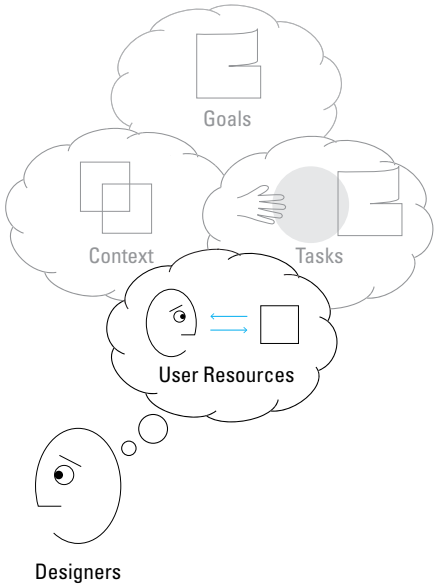
- Concepts (knowledge)
- Methods (skills)
- Rules (informal norms)
- Laws (formal norms)

Attentive designers have in mind user's models of goals, tasks, resources, and context; the conceptual model describes what the user needs to know to employ the resources effectively.

Contexts: Recent, current, anticipated

- Who (actors)
- What (resources)
- Where (location)
- When (time)
- Why (goals)
- How (activities)





Task Domains

- Concepts (knowledge)
- Methods (skills)
- Rules (informal norms)
- Laws (formal norms)

As an example of a conceptual model Johnson + Henderson describe an alarm clock.

The clock stores the **current time** of day,
continually updating it to track the passage of time.

It displays the current time constantly.

Users can set the current time.

Users can set an **alarm** at a **specified time**, or no alarm.

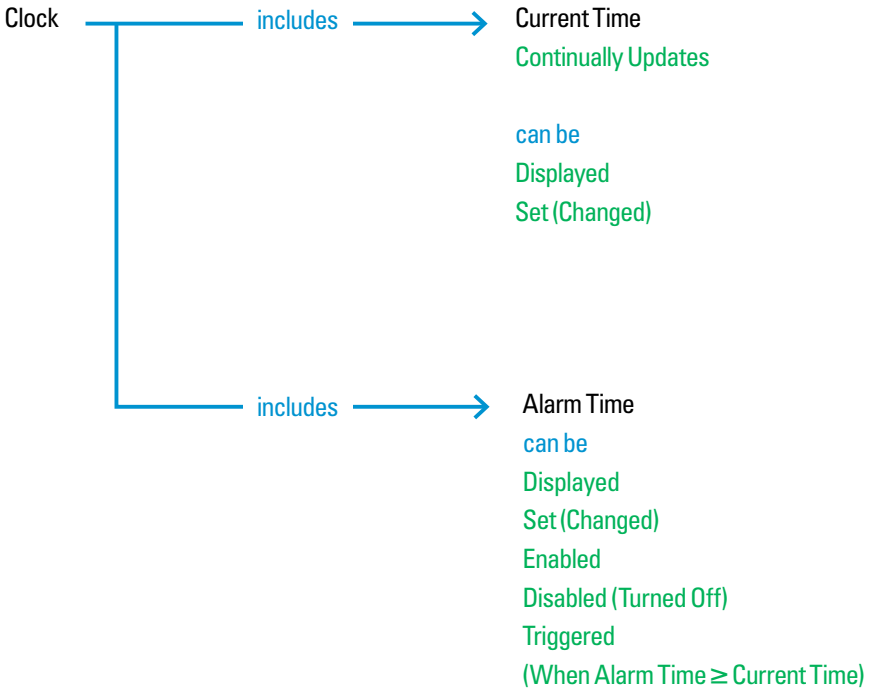
When an alarm is set and the current time equals the set **alarm time**,
the **alarm** is triggered.

Users can turn off an alarm.

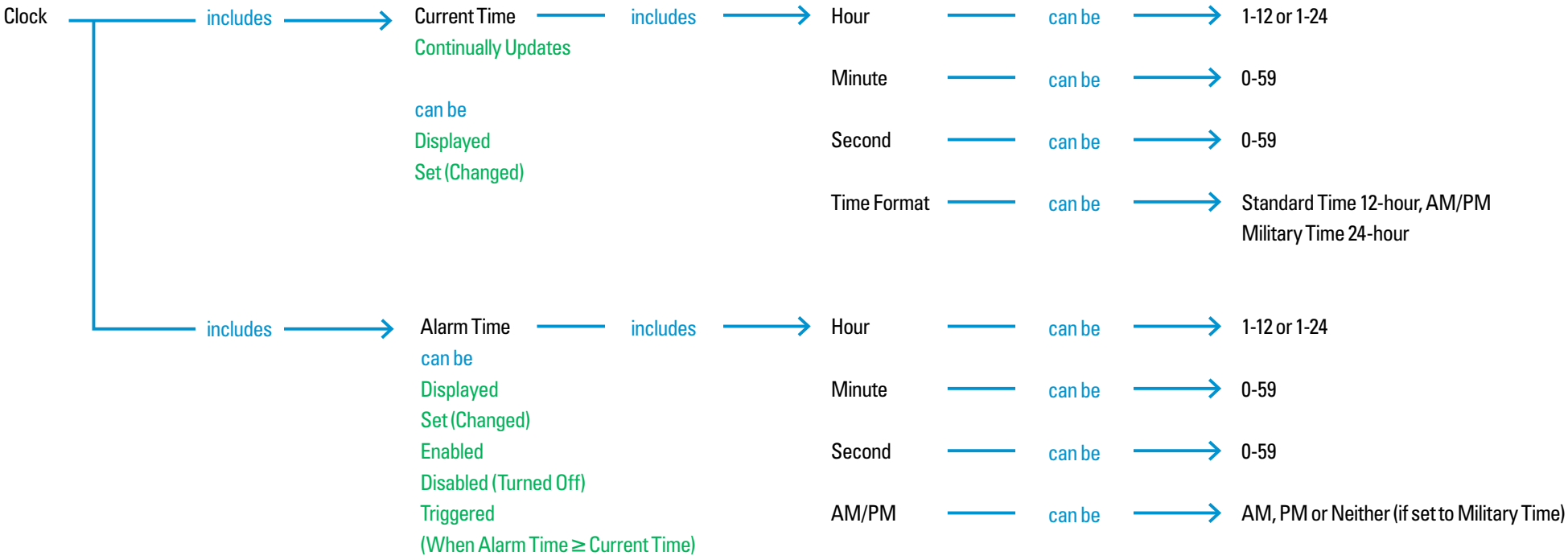
Lists are not models.
Text is often redundant.

Node-link diagrams—
concept maps—
are a more efficient means
of representing
conceptual models

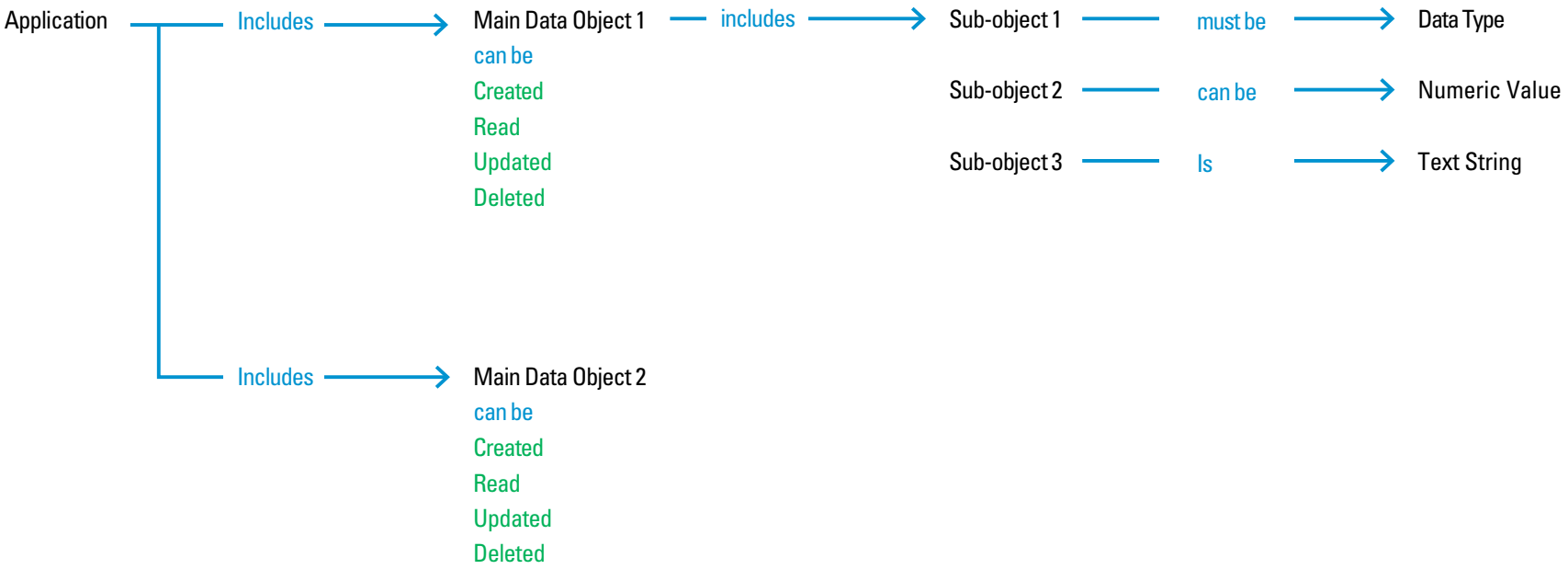
A conceptual model of an alarm clock represented as a concept map.



Conceptual model of an alarm clock with detail added.

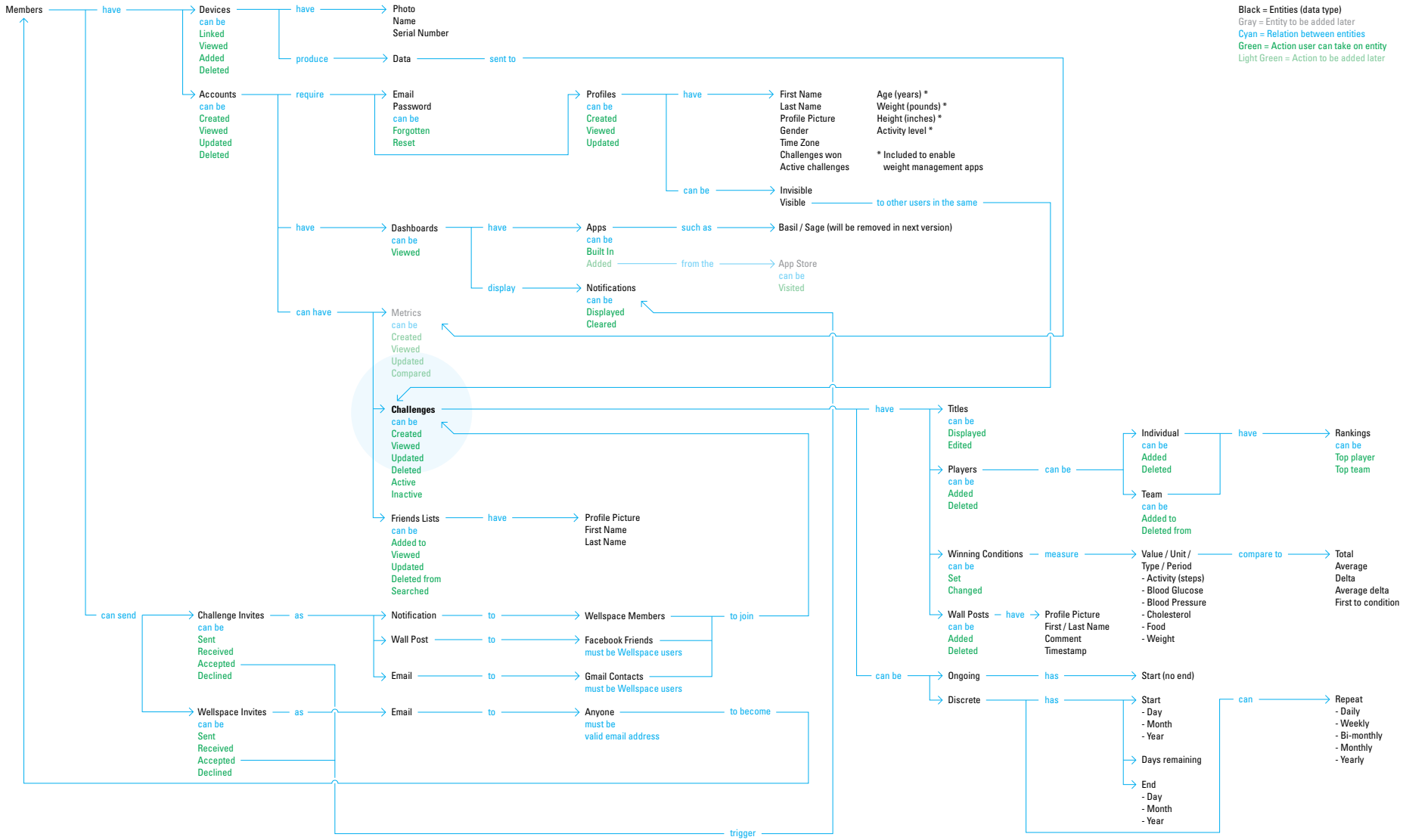


A formalized representation of conceptual models as concept maps.

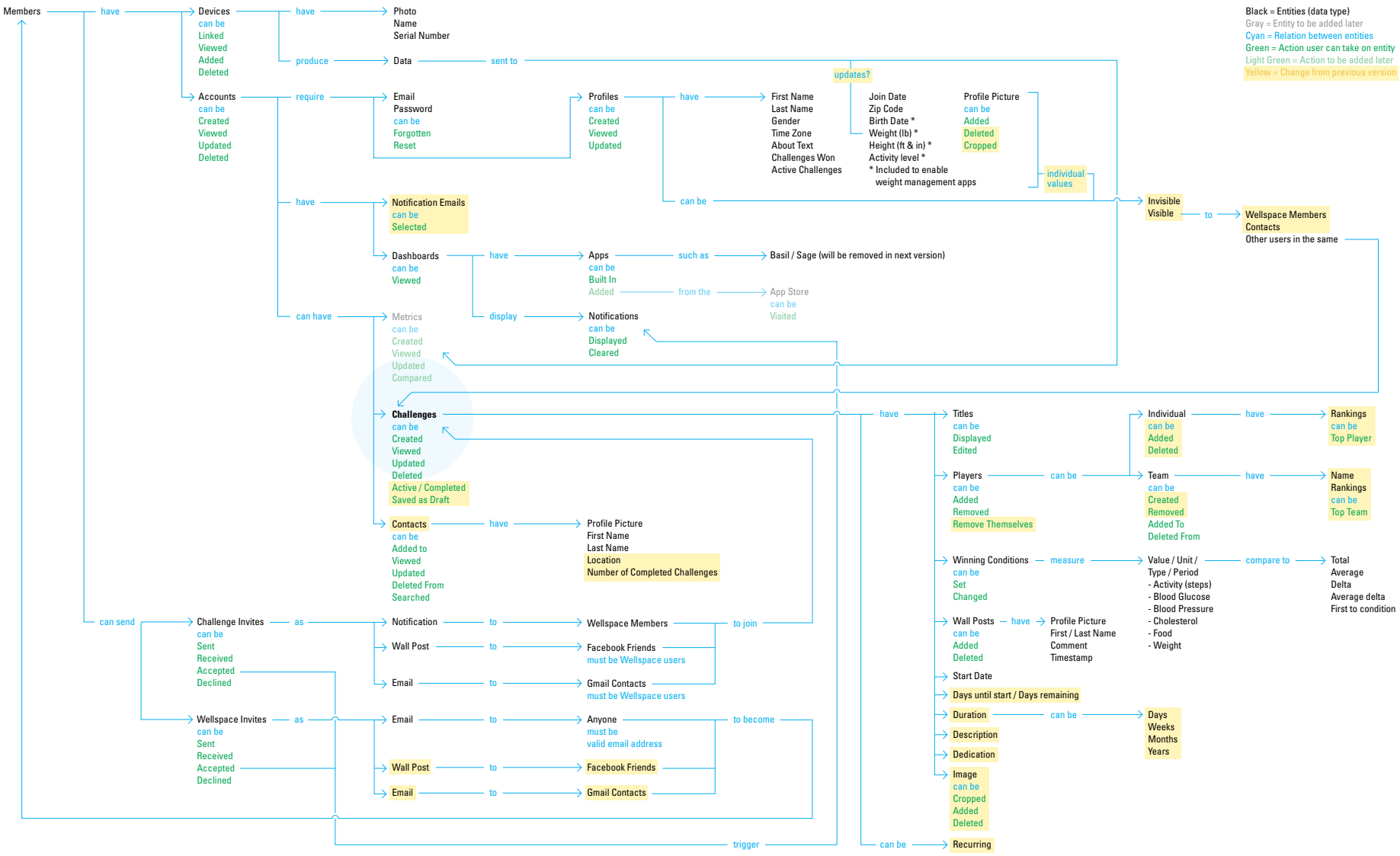


Black = data objects
Blue = relationships
Arrows = data structure
Green = operations
Tinted colors = future

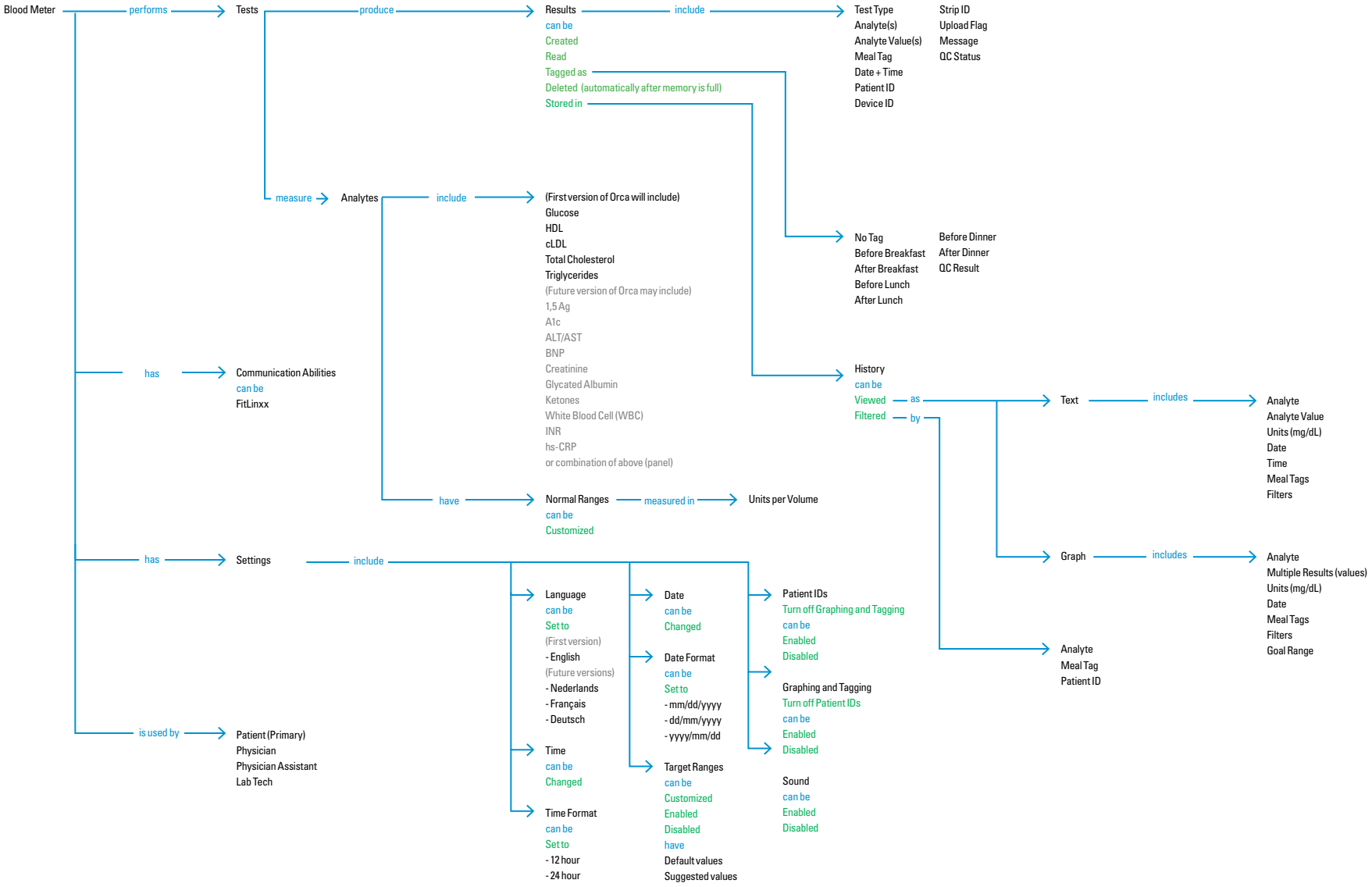
Health challenge example



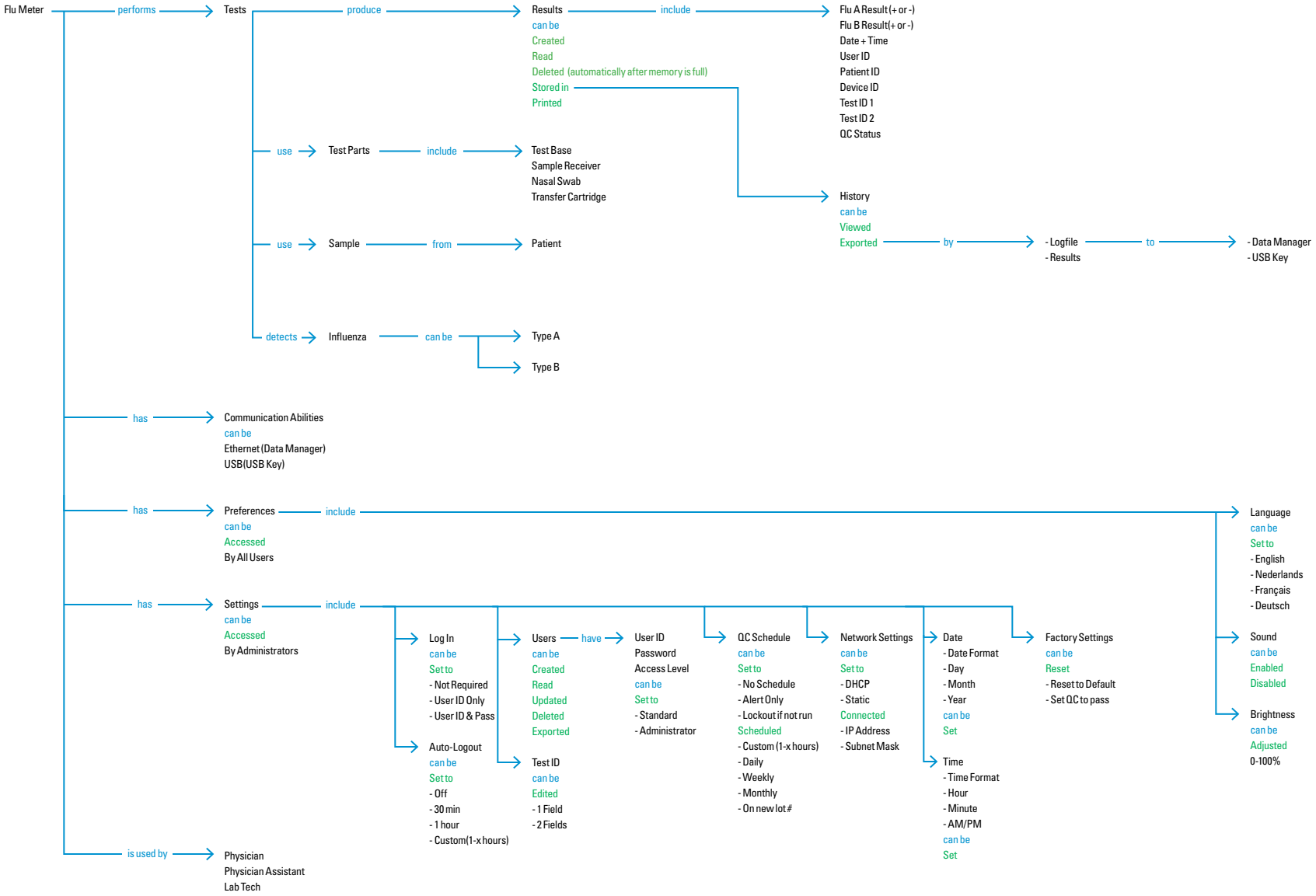
Changes after wireframe design



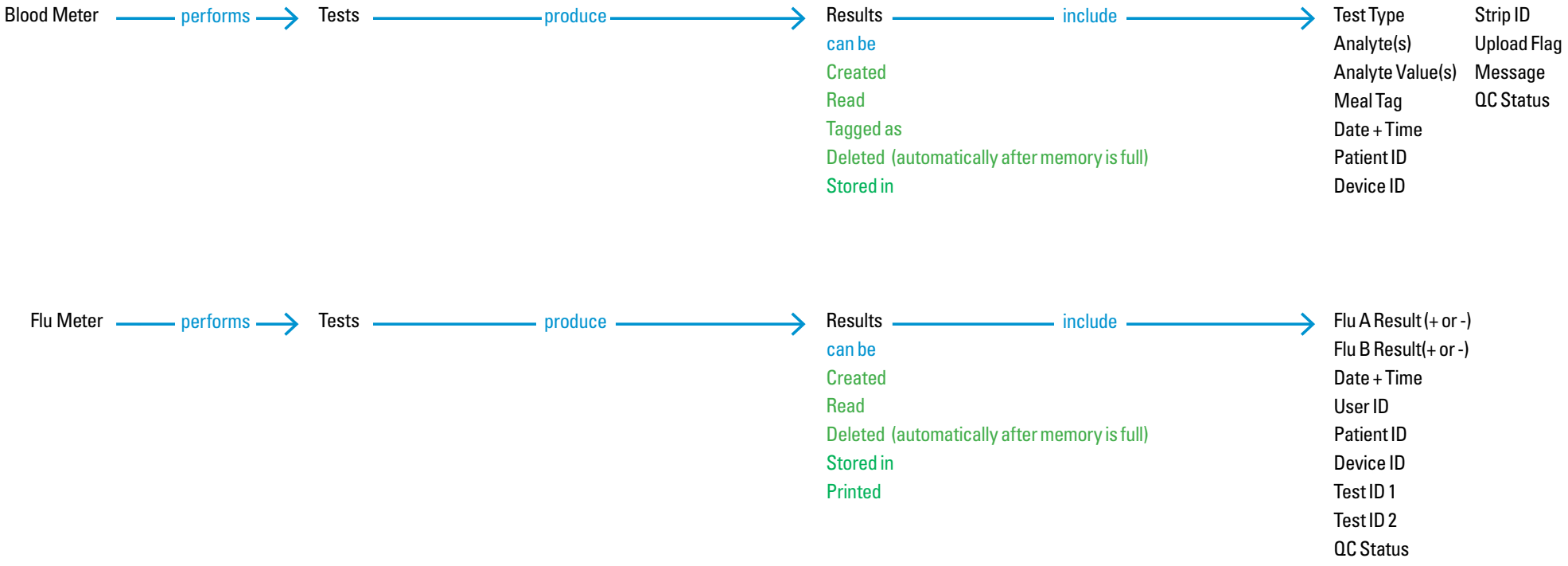
Blood Analyte Meter



Nasal Flu Meter



Meter comparison: Similar primary functions



Different analytes

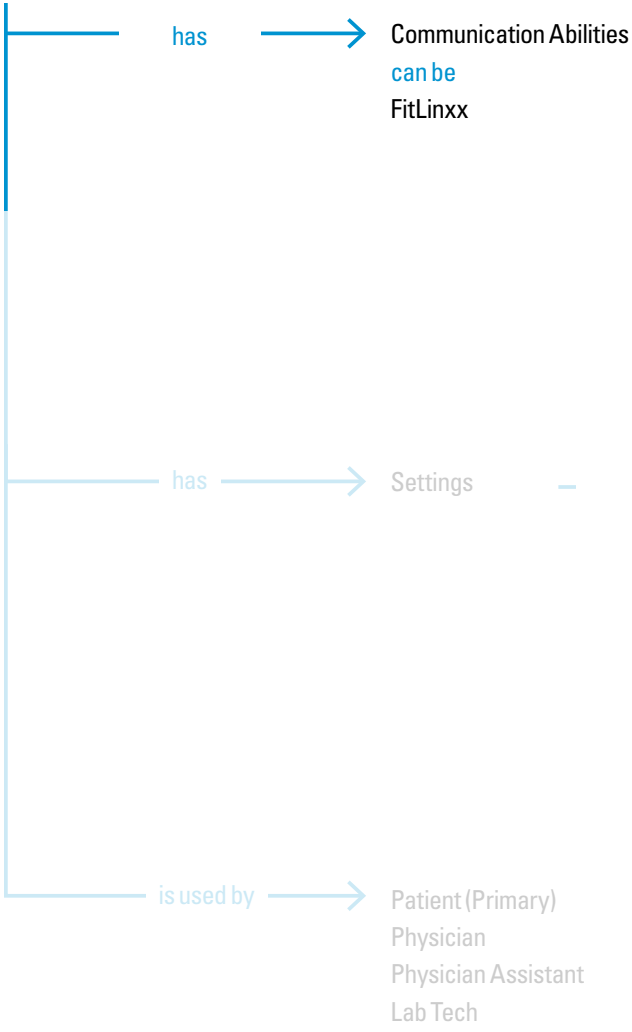
Blood Meter

measure → Analytes — include → (First version of Orca will include)
Glucose
HDL
cLDL
Total Cholesterol
Triglycerides
(Future version of Orca may include)
1,5 Ag
A1c
ALT/AST
BNP
Creatinine
Glycated Albumin
Ketones
White Blood Cell (WBC)
INR
hs-CRP
or combination of above (panel)

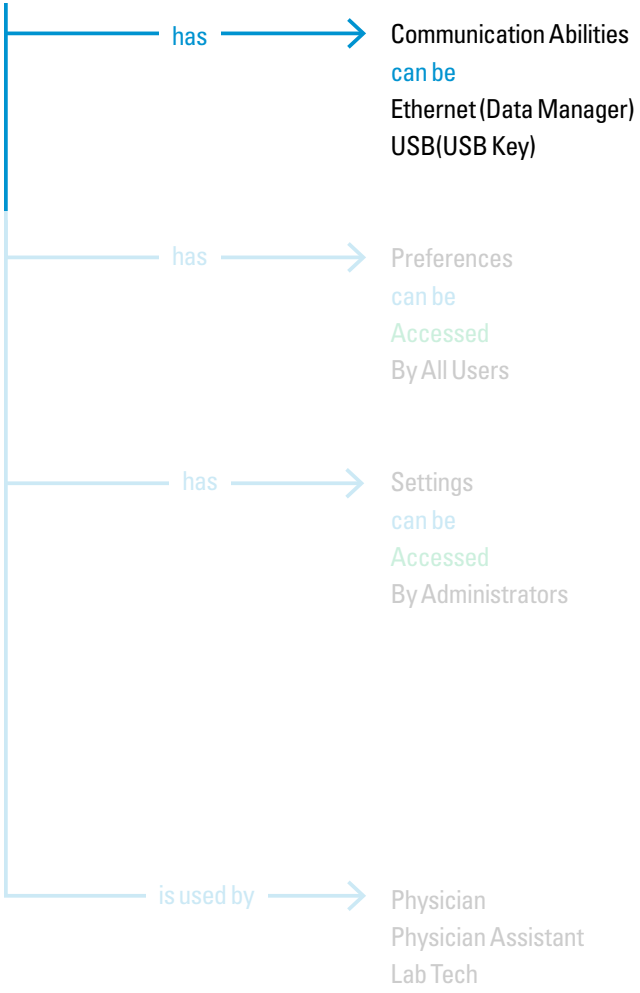
Flu Meter

Different communication protocols

Blood Meter

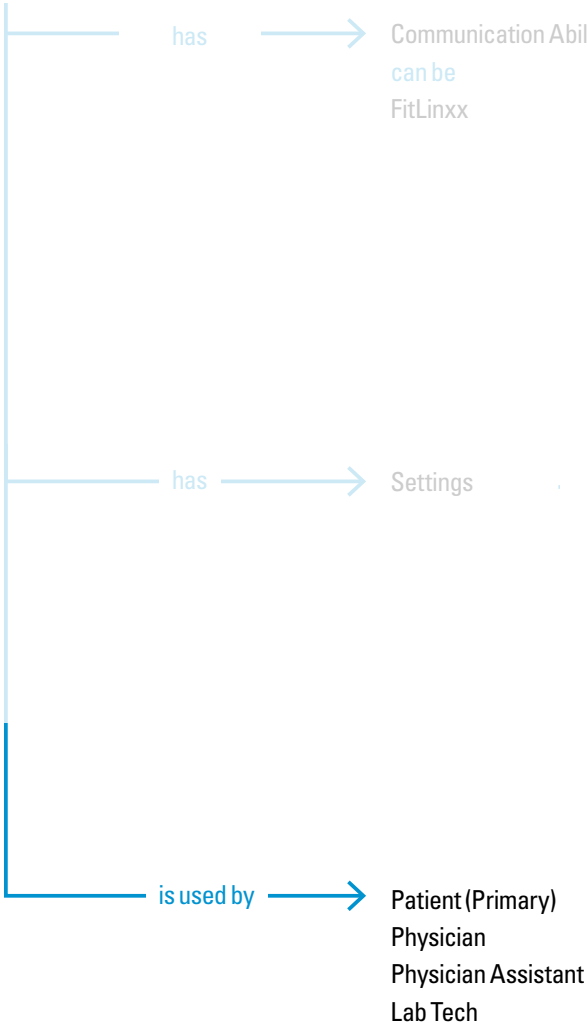


Flu Meter

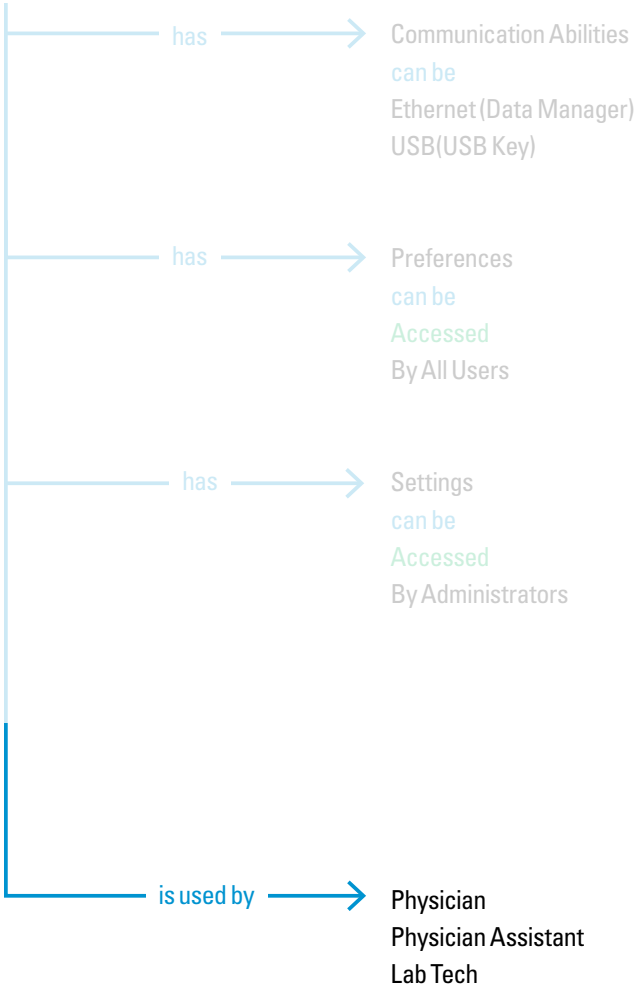


Different user types

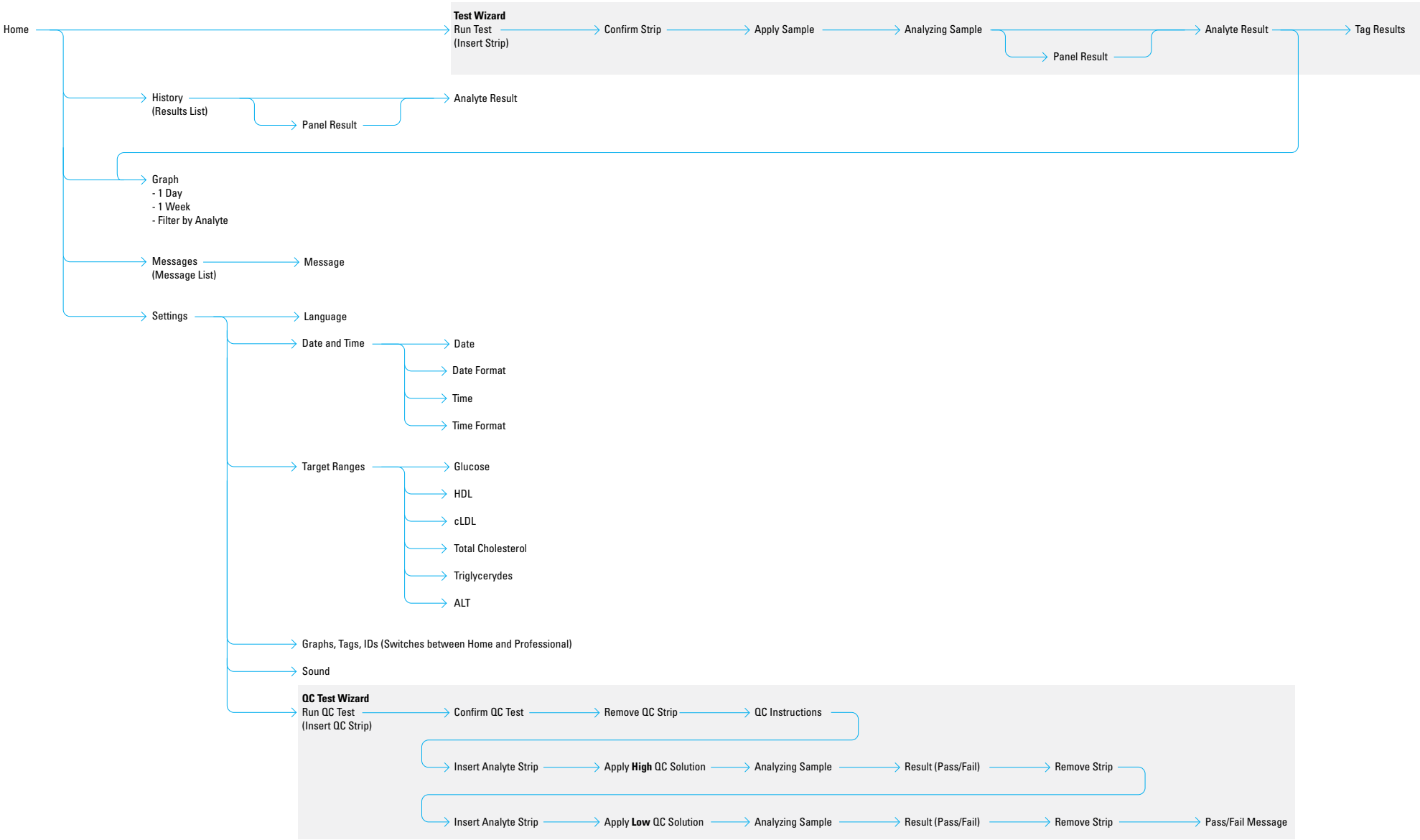
Blood Meter



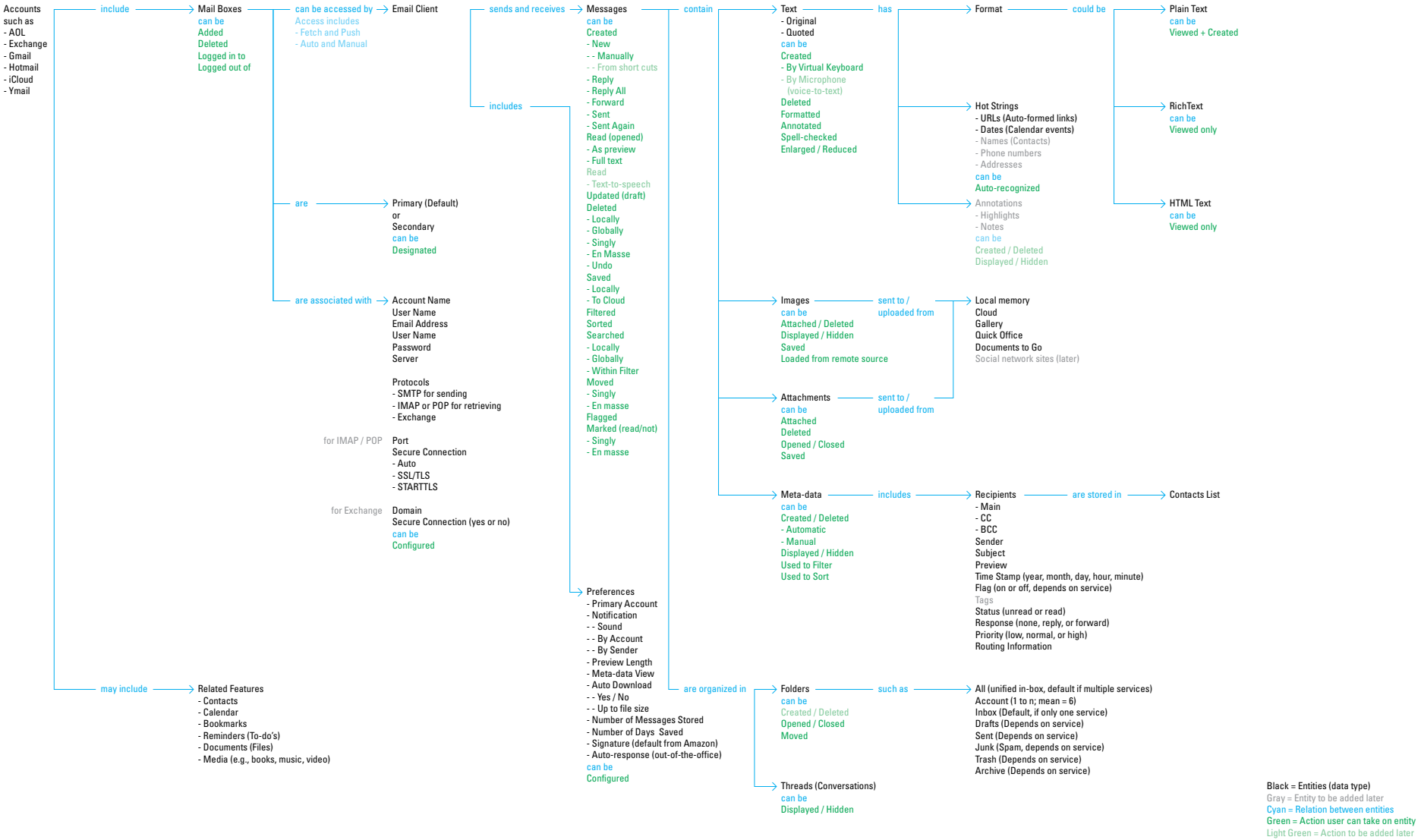
Flu Meter



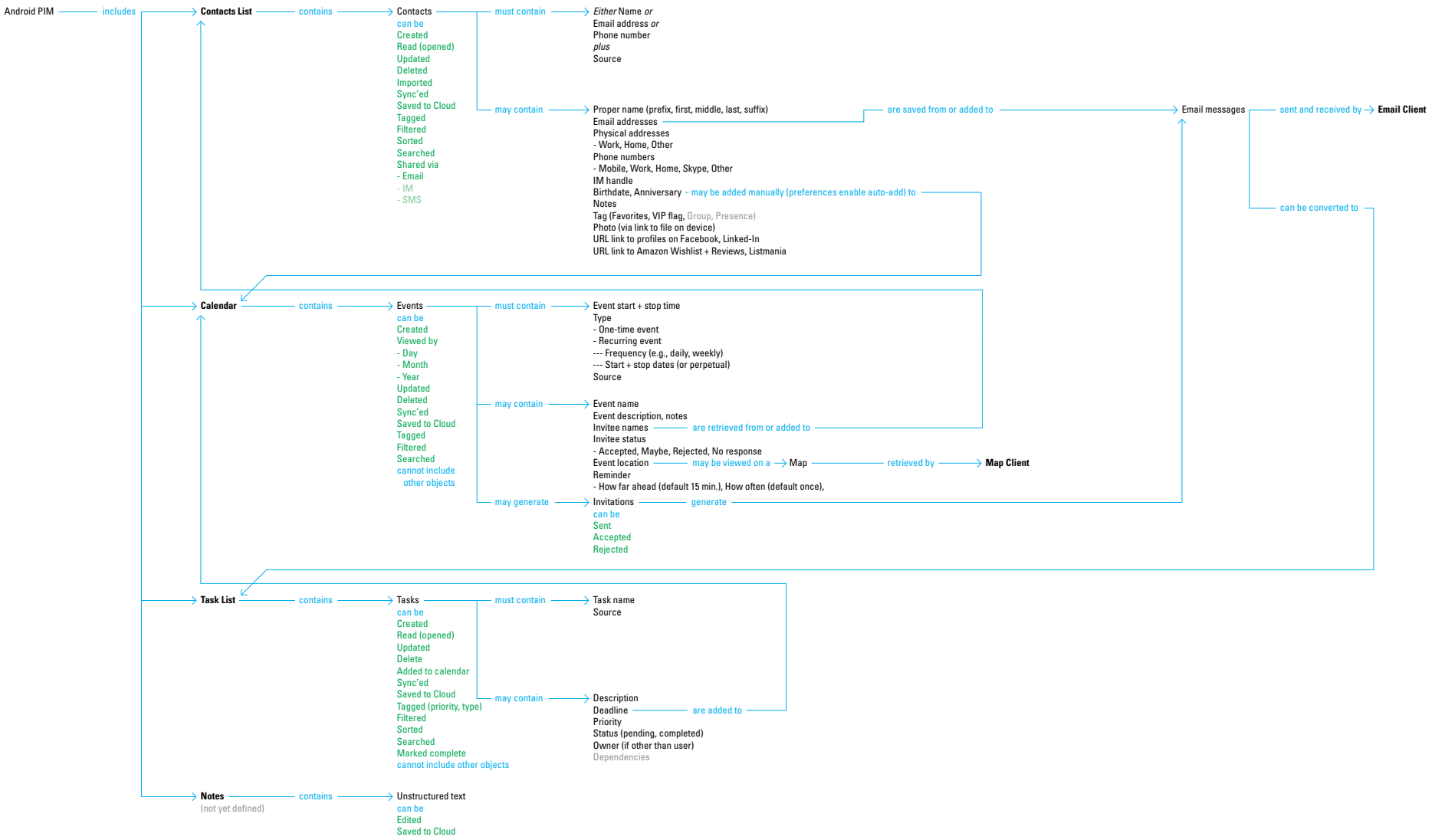
CMs (prior) are not IAs (below)



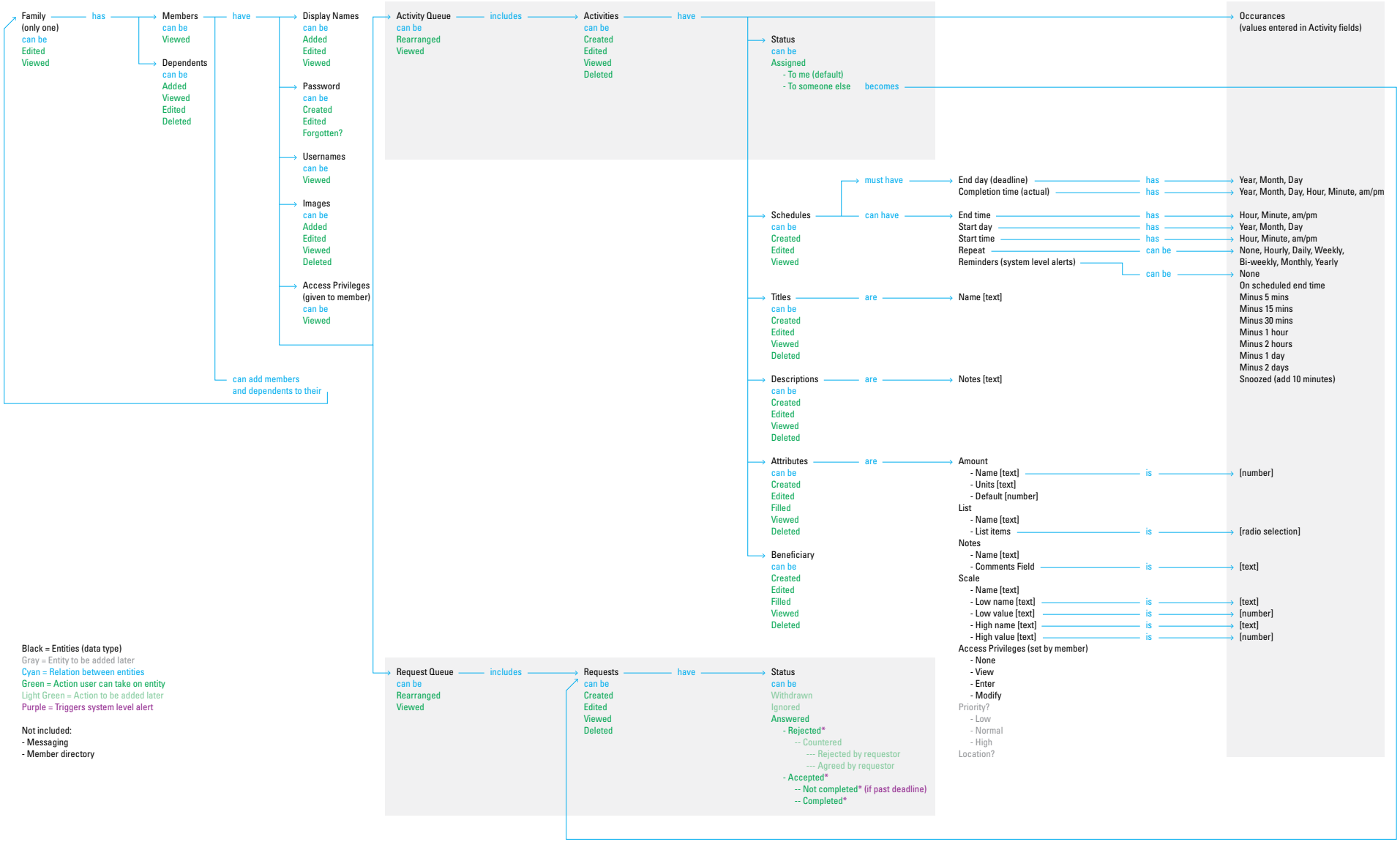
Android, email client



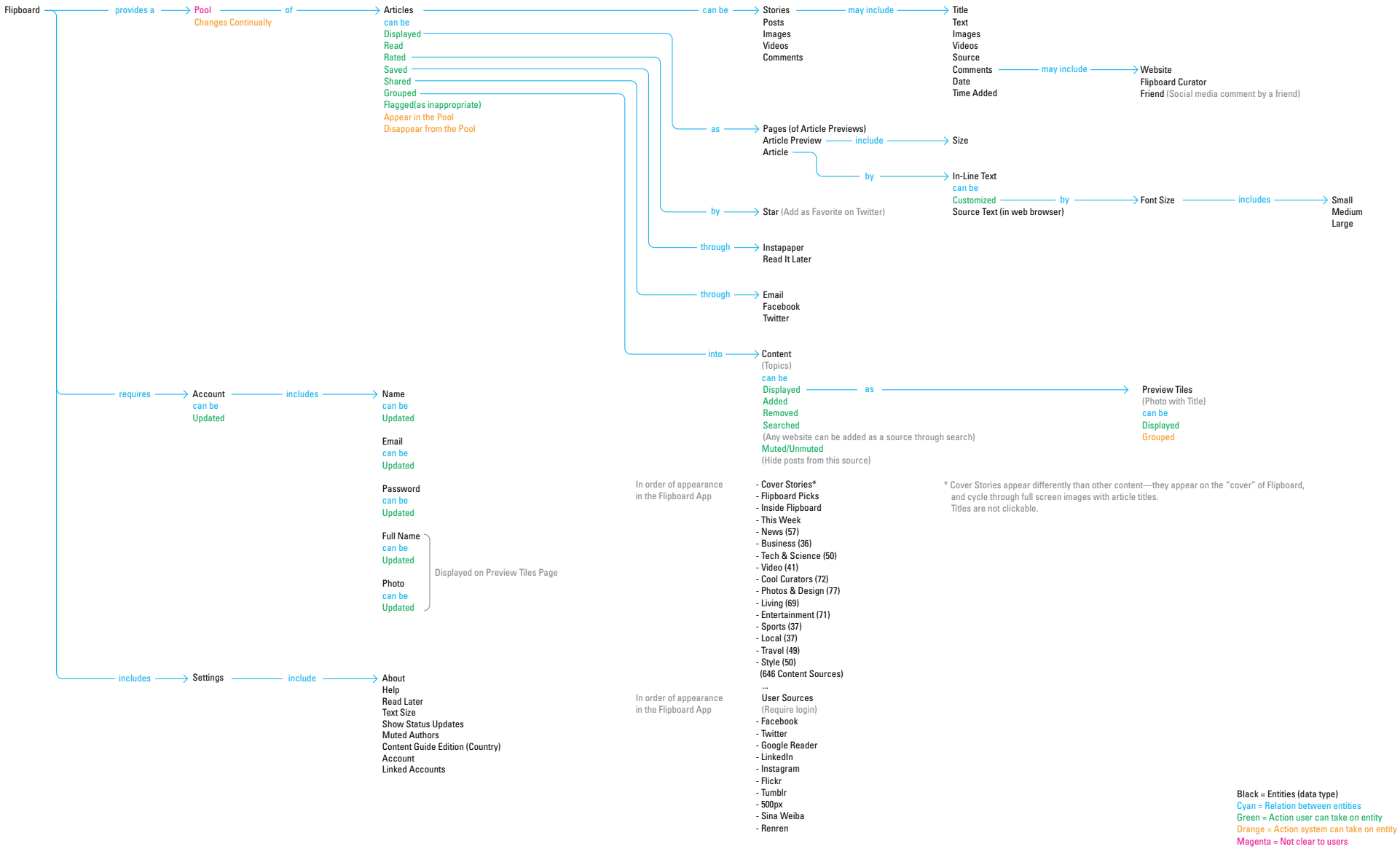
Android, PIM


Android PIM
includes
Contacts List
contains
Contacts
can be
Created
Read (opened)
Updated
Deleted
Imported
Sync'ed
Saved to Cloud
Tagged
Filtered
Sorted
Searched
Shared via
- Email
- IM
- SMS
must contain
Either Name or
Email address or
Phone number
plus
Source
may contain
Proper name (prefix, first, middle, last, suffix)
Email addresses
Physical addresses
- Work, Home, Other
Phone numbers
- Mobile, Work, Home, Skype, Other
IM handle
Birthdate, Anniversary
Notes
Tag (Favorites, VIP flag, Group, Presence)
Photo (via link to file on device)
URL link to profiles on Facebook, Linked-In
URL link to Amazon Wishlist + Reviews, Listmania
are saved from or added to
Email messages
sent and received by
Email Client
can be converted to
may be added manually (preferences enable auto-add) to
Calendar
contains
Events
can be
Created
Viewed by
- Day
- Month
- Year
Updated
Deleted
Sync'ed
Saved to Cloud
Tagged
Filtered
Searched
cannot include
other objects
must contain
Event start + stop time
Type
- One-time event
- Recurring event
--- Frequency (e.g., daily, weekly)
--- Start + stop dates (or perpetual)
Source
may contain
Event name
Event description, notes
Invitee names
are retrieved from or added to
Invitee status
- Accepted, Maybe, Rejected, No response
Event location
may be viewed on a
Map
retrieved by
Map Client
Reminder
- How far ahead (default 15 min.), How often (default once),
may generate
Invitations
generate
can be
Sent
Accepted
Rejected
Task List
contains
Tasks
can be
Created
Read (opened)
Updated
Delete
Added to calendar
Sync'ed
Saved to Cloud
Tagged (priority, type)
Filtered
Sorted
Searched
Marked complete
cannot include other objects
must contain
Task name
Source
may contain
Description
Deadline
are added to
Priority
Status (pending, completed)
Owner (if other than user)
Dependencies
Notes
(not yet defined)
contains
Unstructured text
can be
Edited
Saved to Cloud

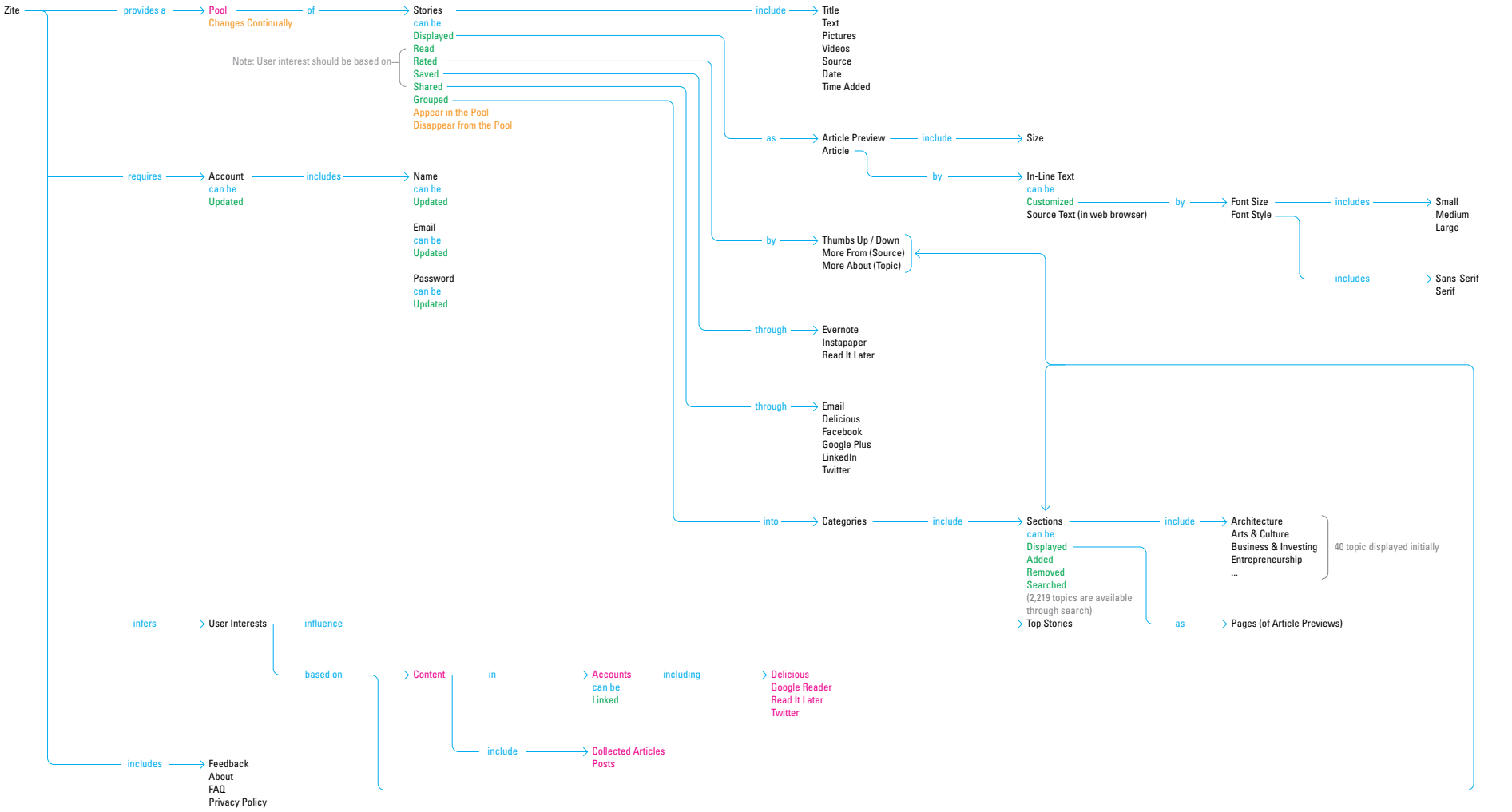
Shared task manager



Flipboard, social magazine

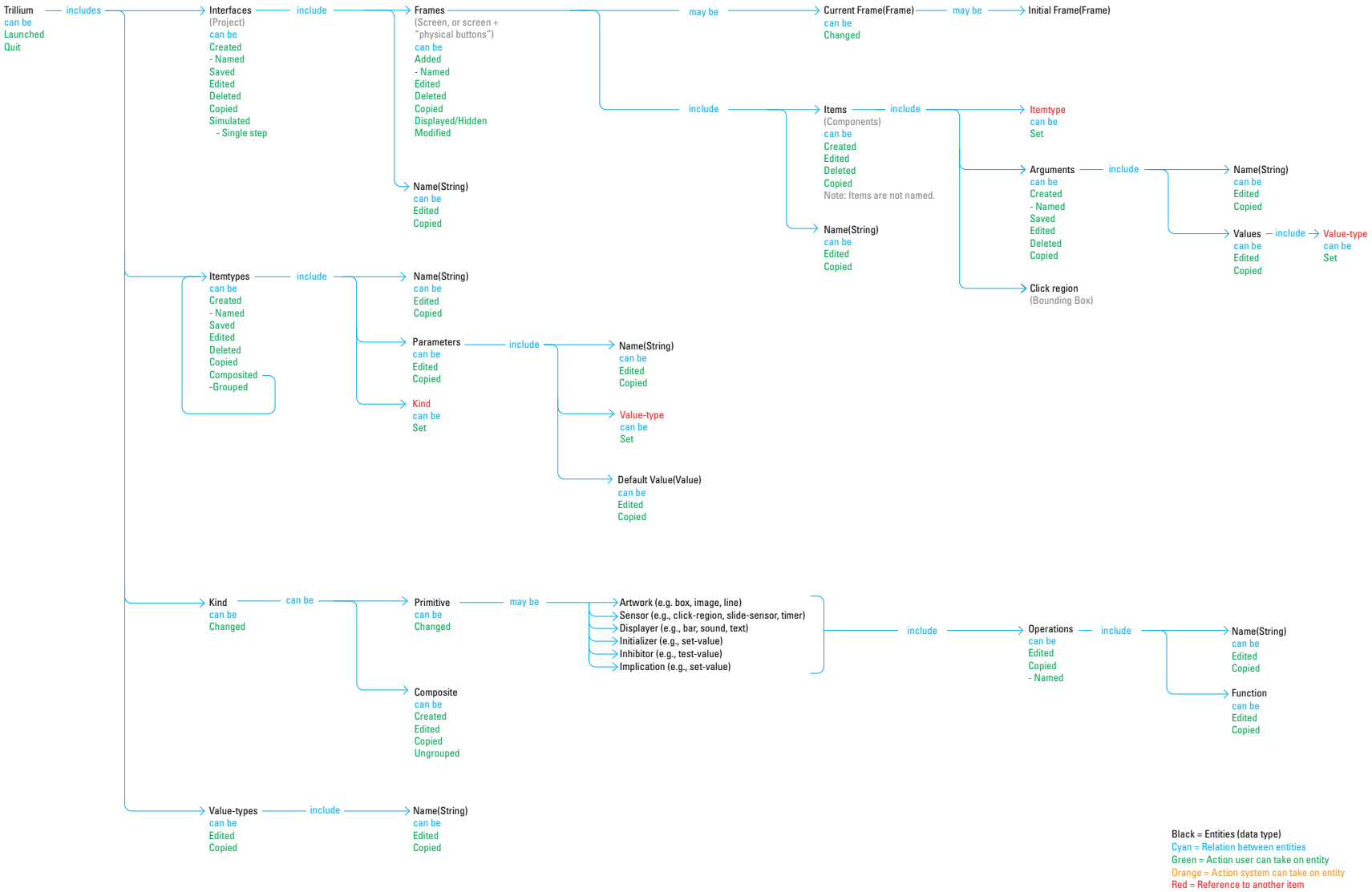


Zite, discovery engine



Black = Entities (data type)
Cyan = Relation between entities
Green = Action user can take on entity
Orange = Action system can take on entity
Magenta = Not clear to users

Trillium, Xerox copier development environment



Teams should agree on a conceptual model—
a definition of what users need to know—
before creating wire-frames and writing code.

The conceptual model will change
during the design and development process.

Developing a standard form for conceptual models makes them easier to understand and make—and easier to learn.

A standard form makes practice more efficient and builds a transferable body of knowledge.

Special thanks to
Terry Irwin
Jodi Forlizzi
Austin Henderson
Jeff Johnson
Michael Gallagher

hugh@dubberly.com
415 648 9799

Presentation posted at
www.dubberly.com/presentations/cmu_systems.pdf