Twinlist: Overview and general implementation description

Tiffany Chao (tchao@cs.umd.edu)
Catherine Plaisant (plaisant@cs.umd.edu)
Ben Shneiderman (ben@cs.umd.edu)

Human-Computer Interaction Lab
University of Maryland, College Park, MD

January 11th, 2012

1 Introduction

Medication reconciliation can help prevent adverse health outcomes, but the process is nontrivial. Here we focus on the user interface that clinicians might use to compare and merge two separate lists of medications. Similarities and differences between the lists need to be identified; decisions need to be made as to which medications should be continued and which ones should be stopped. We describe a novel prototype called Twinlist that uses animation and spatial layout to reveal similarities, and simple interactive controls to facilitate “accept” or “reject” decisions. We also suggest extensions and modifications that may be useful for widespread adoption (provided as “notes” in a smaller font size).

Twinlist was designed as an inspirational standalone prototype. If adopted, it would become part of the user interface of a larger system (e.g. a CPOE or EHR system). Here we will refer to that interface as the “outer” interface.

This document assumes that the reader is already familiar with Twinlist’s video demonstration.

2 The reconciliation problem

The entire process of medication reconciliation is a complex collaborative process in which many things can go wrong: the patients may not recall what medications they are taking (or may even be unconscious); the information may not be recorded properly and include a lot of unreported uncertainty (e.g. about dosage); the record of past medication orders may not be complete or not even be accessible at all; not all sources of medication orders for the patient may be known (e.g. they may have consulted a specialist on their own), etc. Eventually the physician is presented with multiple lists of medications from different sources that must reconciled into a single medication regimen. **Our focus has been on this last step of the**

process: facilitating the task of reviewing and sorting the medications that need to be continued from those that need to be stopped. Our interviews of physicians suggest that they do not have adequate tools or a well-described workflow to perform medication reconciliation.

4 Assumptions about the data

Twinlist focuses on user interaction, and is thus deliberately independent of the similarity pre-processing algorithm. Our prototype assumes that the similarity information will be generated by a separate algorithm; such an algorithm has been developed by our SHARPC colleagues at the University of Texas in Houston. Please contact Jorge R. Herskovic (Jorge.R.Herskovic @ uth.tmc.edu) to gain access to the algorithm, or read their publication "Automated medication reconciliation and complexity of care transitions", Bozzo Silva, Bernstam, Markowitz, Johnson, Zhang and Herskovic, AMIA 2011- http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3243161/ .

Note: if no such existing algorithm can be developed or acquired, one could still use Twinlist by first performing a straight string comparison between drugs. Exact matches would be "identical", drug name match only would be classified as a "similarity", and all other drugs would appear as "unique". This would miss a lot of similarity information, but may be a good way to start implementing Twinlist.

Twinlist defines medications as a set of five attributes (name, route, frequency, dosage, and purpose), and assigns each a unique numeric id. List items are thus identical if all five attributes are exactly the same, similar if between one and four attributes match (and share an active ingredient), and unique if they are neither identical or similar to any medication. Drug-drug similarities will generally be between pairs of drugs in separate lists, but the current prototype can also handle one-to-many or many-to-many similarities, including similarities between drugs in the same list.

5 Description of the interface

Twinlist's interface consists of three parts (Figure 1): the header (top), the list viewer (center), and the item detail (bottom). The dark gray header is the main navigation. The list viewer is where users may interactively accept or reject medications. The item detail displays all available information about the medication that the user is currently hovering over -- including what the drug is used for, which the list viewer omits to save valuable screen space.

Note: While larger screens may allow for alternative layouts, the current prototype was designed for a 1024 x 768 pixel screen and used a fairly large font so we could demonstrate it easily. The main menu is thus built into the page header (top), and the item detail occupies a separate stripe below (a fixed position, as it must always be visible). This layout would extend gracefully to larger resolutions as-is, but designers could also consider moving menus and/or the details on either side of the screen. We found that tooltips were not practical to show drug details because that information can be very long and we could not eliminate all cases of occlusion that would hide similar drugs that users needed to see.

Interaction begins with an introductory window that swipes down from the top of the screen. Once the user closes the introduction window, the two lists to be reconciled sweep in from the top left corner and settle into two distinct columns.

Note: Most likely, this introduction is only useful in our standalone demonstration. Real systems should integrate help and instructions in the "outer" (containing) interface; within that context, it may be best to simply have the two lists appear from the start. Note also that the current layout does not display the patient information -- if the "outer" interface encapsulating Twinlist does not display that information, then the the list reconciliation interface should.

Twinlist's sample datasets illustrate possible hospital discharge scenarios of use, in which hospital physicians must reconcile **Intake** and **Hospital** lists. The **Intake** list (left) corresponds to what the patient reported taking before he was hospitalized (as recorded during the intake interview); the **Hospital** (right) lists what the patient took during his stay. Different use cases will naturally require different list names.

Note: If more than 2 lists need to be reconciled, you could repeat the 2-list reconciliation step multiple times: the left list could then be cast as the "current" or "EHR" list, which would be merged with the right list -- the "new" or "foreign" list. If the lists are short enough, the n-list problem could simply be transformed into a 2-list problem by combining n - 1 lists.

Clicking "Compare lists" triggers an explanatory animation that reveals similarity information: identical drugs first move to the center, one set at a time (Figure 2); then unique drugs move to the side, first to the left for the drugs unique to the intake list, then to the right for the drugs unique to the hospital (Figure 3); similar drugs are aligned all at once (Figure 4); finally the display is compacted to save vertical space (Figure 5). Golden-yellow highlights are added to indicate the differences among similar drugs.

Note: While this animation is very attractive and helps explain the final layout of the drugs on the screen, it is possible that some users may prefer to skip the animation if they already familiar with layout. User testing will inform the final design, but it is likely that an option to accelerate the animation or to skip it entirely should be offered.

Once the animation completes, users may click to accept or reject medications as they see fit. A left-click accepts the drug, a right-click rejects it, and double-click to undo and return to the "undecided" state. Practically clicking on a medication toggles through three states: undecided, accepted, and rejected. Left-click cycles undecided → accepted → rejected; right-click cycles undecided → rejected → accepted. Hence changing to a new state is always possible with one left- or right-click.

Note: Further restricting or tweaking toggle behavior may be preferable, especially if the right-click is needed or desirable for other context-specific actions (e.g. if modifying the order is allowed). For example, small ephemeral buttons for accept and reject could appear below a selected drug. Tablet or phone-based implementations might consider using a tap to select and a swipe gesture to the left to reject (as it is becoming a standard for [delete] in many apps).

Explicit accept and reject buttons beneath column headers provide a convenient way to accept or reject entire columns when appropriate. It is unclear whether column-accept and column-reject should overwrite previous decisions; In the current implementation we decided to errs on the side of caution and only apply the command to the medications that remained "undecided".

Note: We decided against manual multi-selection (e.g. using the CTRL key) because the current implementation only requires a simple click to make a decision, thereby eliminating the need to first group drugs before specifying the decision.

When users have acted on every medication, clicking **confirm choices** will finalize their decisions, transitioning to a final summary layout (Figure 6) where they may **sign off** on their reconciled list.

Note: Here again, the reconciliation interface will depend on the "outer" interface and scenario of use. In a discharge setting, for example, it could conceivably feed into patient discharge forms.

If not all the drugs have been taken care of a warning dialog box appears, indicating the number of drugs that remain to be reviewed.

Note: Alternatively the undecided drugs could be listed in the dialog box and an option be given to accept or reject them all. We decided against this, thinking that it would always be better to review the drugs in the original lists so that the similarity information would be visible.

Advanced features are hidden by default, accessible only via **show options** (Figure 7). Here users may customize the list viewer to suit special needs: they may sort by name or dosage, group by route (e.g. to make sure no IVs remain) or by therapeutic intent, and/or filter on any fragment of a medication's name.

Note/limitation: *The current prototype assumes that medications only have one such "therapeutic intent." Clearly, this is not always the case. n-group affiliation, however, remains a worthwhile extension. The obvious (but not particularly scalable) extension would require duplicating the medication n times, one for every group. Another option would be to introduce group-wise reconciliation: first act on all heart medications, then on all pain medications, and so on.*

Users can also choose to hide/remove accepted and rejected medications as they are being acted upon to regain more screen space to show the remaining drugs. This option might be particularly useful for very large lists.

Font size is controllable using the standard web browser controls (CTRL-+ and CTRL--). If those controls are not frequently used in the outer interface, adding explicit buttons might be required.

All options should be saved by the "outer" interface in the user profile.

The current prototype also allows users to replay any stage of the list reconciliation animation. This was very important for us to demonstrate the interface and may still be useful for new users (to replay the animation steps one at a time), but those controls should not be the most prominent in the option panel as they are less important.

6 Visual design

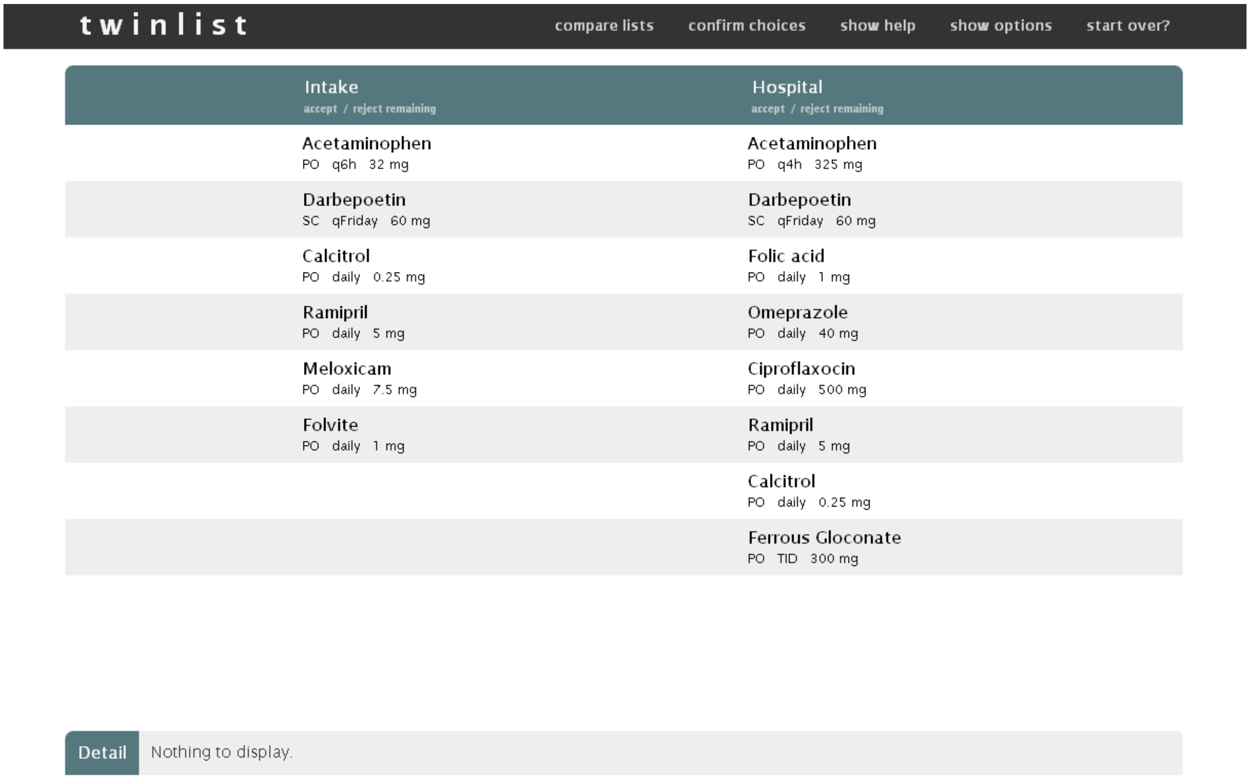
Design can either break or make the interface. Line, color, texture, form, space -- all of these can make things appear simple and understandable, or overly complex. Meaningless colors and shapes are wasted opportunities and confusing.

In Twinlist particular attention was paid to visual design. Solid colors, used sparingly, define the interface (Figure 8): dark gray anchors the header to the top of the page; bright white lends an illusion of spaciousness; blue-green provides richness and life; golden-yellow highlights important differences between related items; vibrant yellow-green lets users know which drugs have been accepted at a glance. All click-able objects further provide feedback on mouseover: the list viewer in particular uses a slight "nudge" effect to group related medications, exploiting the immediacy of motion and the Gestalt principle of common fate to guide visual queries.

Note: Color schemes and interaction cues need to be consistent with those used in the "outer" application. No matter what, however, there should sufficient feedback to keep users informed as the state of each drug. (Better yet, keep them engaged!)

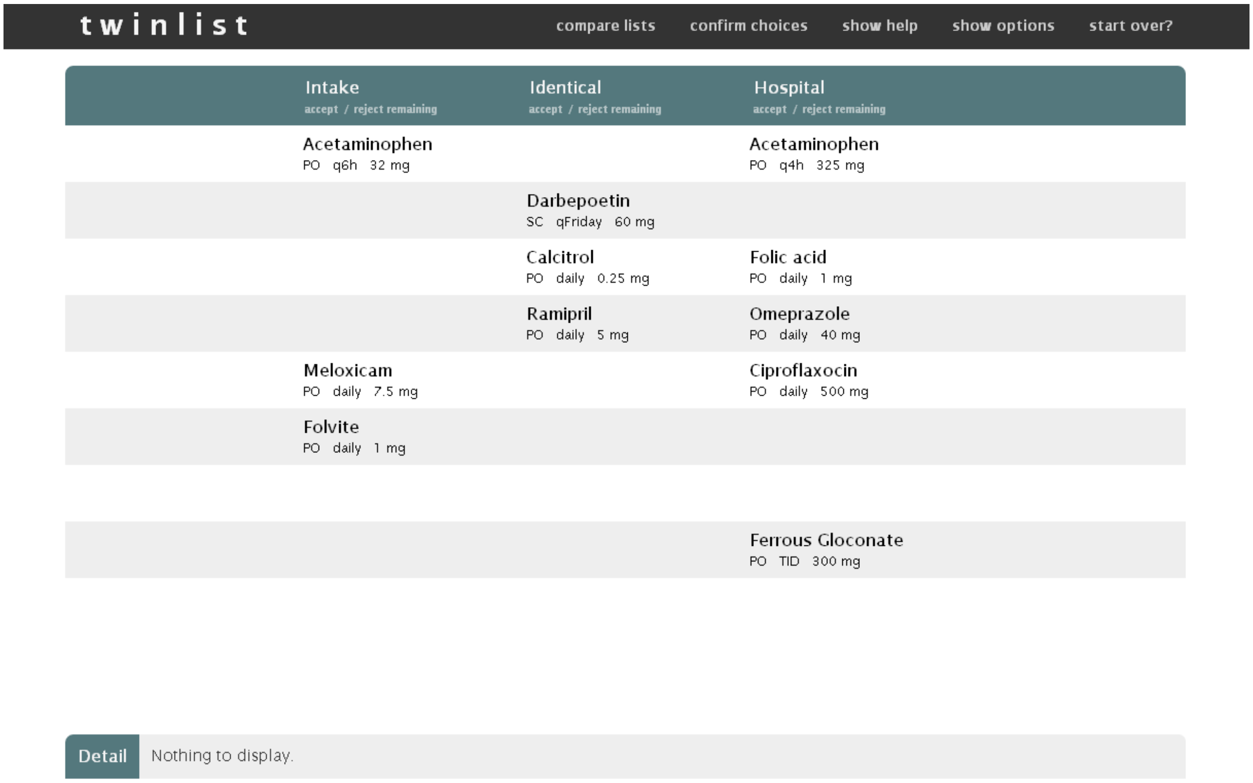
7 Animation design

We broke a complex overall transition into separate steps, allocating approximately 1 second per step, with up to 750 ms to properly punctuate separate steps. Multi-step animations seem to benefit from a unifying narrative -- even something as simple as "all of the squares are going to change from blue to gray". Take time to choreograph in a natural sequence of events. Carefully direct user attention to lessen the burden of comprehension.



Intake	Hospital
Acetaminophen PO q6h 32 mg	Acetaminophen PO q4h 325 mg
Darbepoetin SC qFriday 60 mg	Darbepoetin SC qFriday 60 mg
Calcitrol PO daily 0.25 mg	Folic acid PO daily 1 mg
Ramipril PO daily 5 mg	Omeprazole PO daily 40 mg
Meloxicam PO daily 7.5 mg	Ciproflaxocin PO daily 500 mg
Folvite PO daily 1 mg	Ramipril PO daily 5 mg
	Calcitrol PO daily 0.25 mg
	Ferrous Gloconate PO TID 300 mg

Figure 1. Separate lists. The two lists start as two separate columns, **Intake** and **Hospital**. List items are unsorted by default.

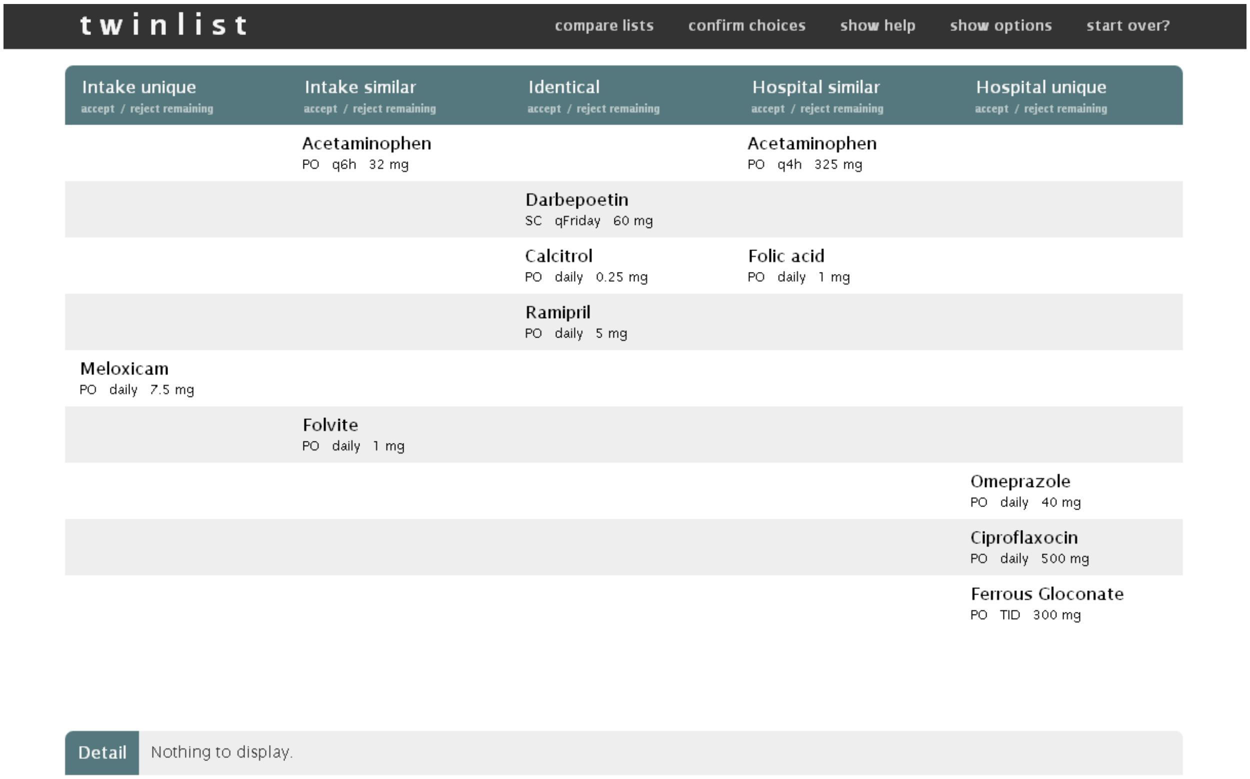

twinlist
compare lists
confirm choices
show help
show options
start over?
accept / reject remaining
accept / reject remaining
accept / reject remaining
Detail

Intake	Identical	Hospital
Acetaminophen PO q6h 32 mg		Acetaminophen PO q4h 325 mg
	Darbepoetin SC qFriday 60 mg	
	Calcitrol PO daily 0.25 mg	Folic acid PO daily 1 mg
	Ramipril PO daily 5 mg	Omeprazole PO daily 40 mg
Meloxicam PO daily 7.5 mg		Ciprofloxacin PO daily 500 mg
Folvite PO daily 1 mg		
		Ferrous Gloconate PO TID 300 mg

Detail: Nothing to display.

Figure 2. Identical items converge. Items from the same set animate in unison; the sets themselves animate sequentially. Each set waits for its turn: items from the first set fly to the center and converge, then items from the second set fly to the center and converge, and so on. Note that the position of the item from the first list takes precedence (compare Ramipril's position above to its position in Figure 1).

Note: Animating identical sets sequentially helps users track individual items.

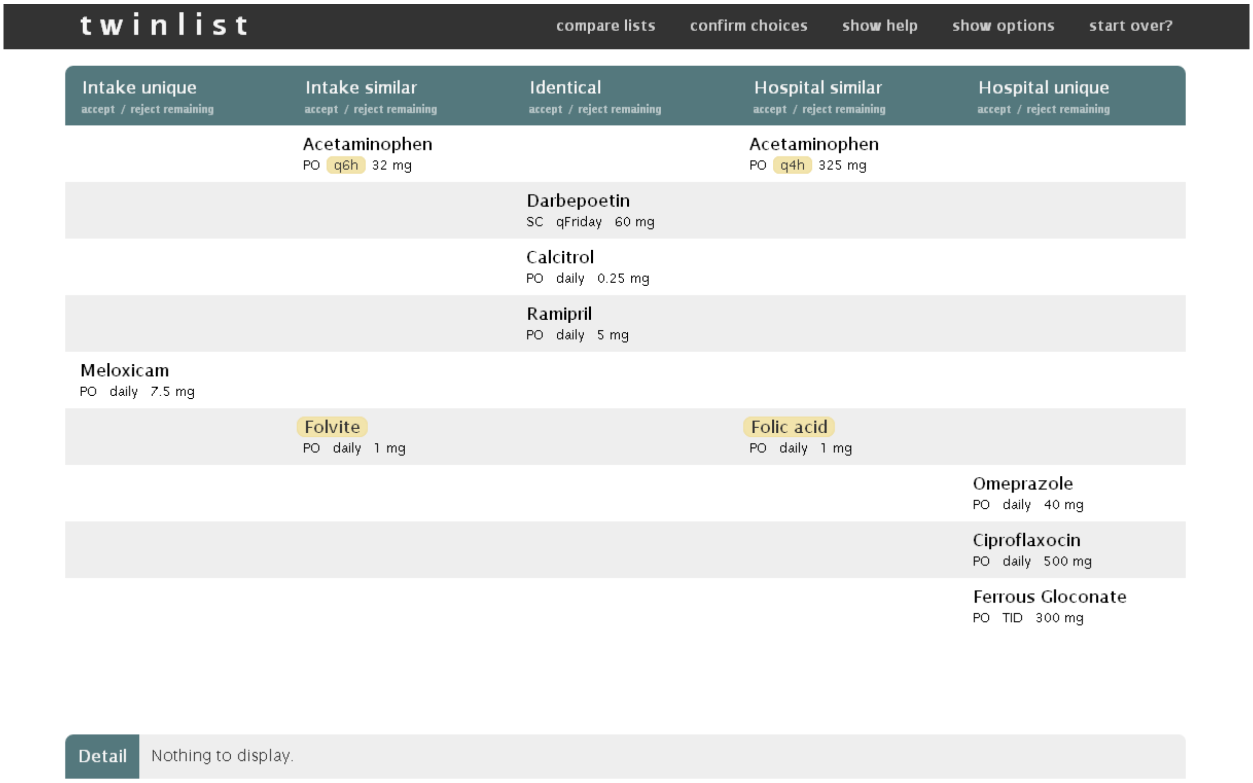


Intake unique accept / reject remaining	Intake similar accept / reject remaining	Identical accept / reject remaining	Hospital similar accept / reject remaining	Hospital unique accept / reject remaining
	Acetaminophen PO q6h 32 mg		Acetaminophen PO q4h 325 mg	
		Darbepoetin SC qFriday 60 mg		
		Calcitrol PO daily 0.25 mg	Folic acid PO daily 1 mg	
		Ramipril PO daily 5 mg		
Meloxicam PO daily 7.5 mg				
	Folvite PO daily 1 mg			
				Omeprazole PO daily 40 mg
				Ciprofloxacin PO daily 500 mg
				Ferrous Gloconate PO TID 300 mg

Detail Nothing to display.

Figure 3. Separate unique items. Medications that are unique to either list fly to the leftmost and rightmost columns. This transition is a two-part animation: first all items that are unique to **Intake** fly to the left, then all items that are unique to **Hospital** fly to the right.

Note: The message here is: "these things are different". Animating one side at a time emphasizes this separation; individual item tracking is less important for this phase.

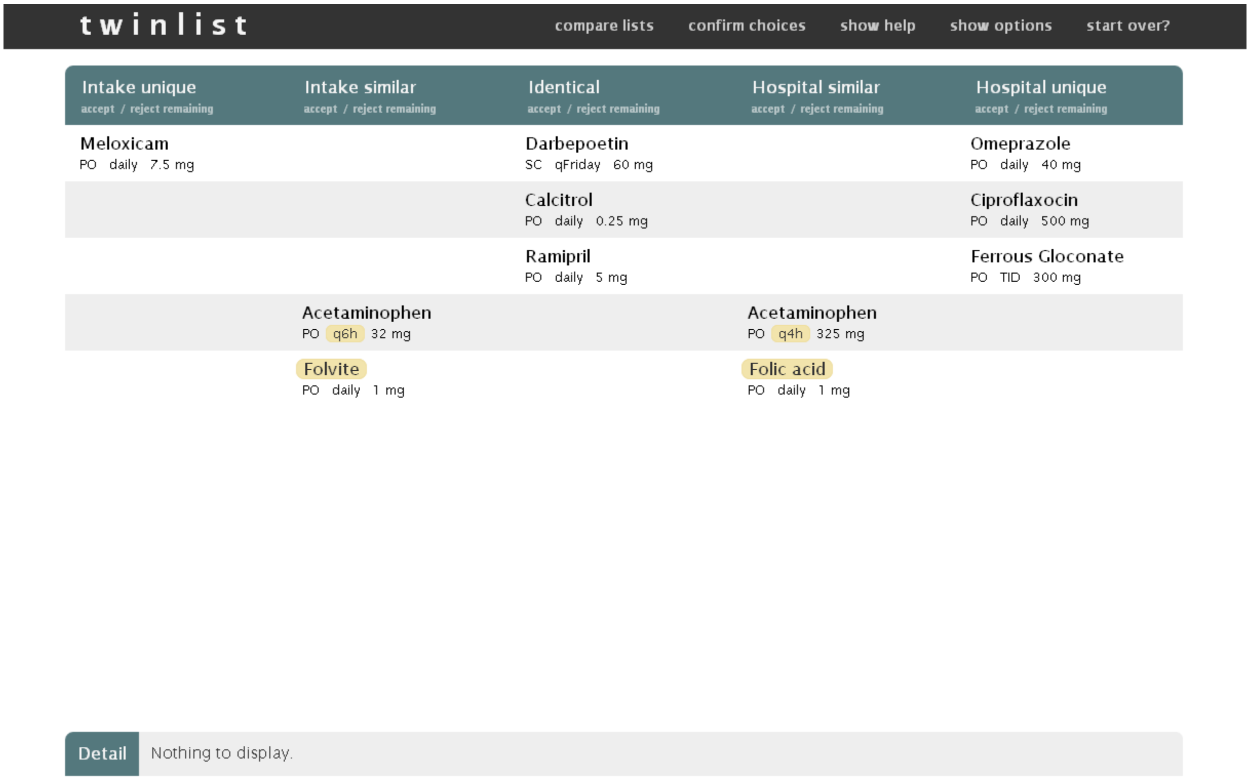

twinlist
compare lists
confirm choices
show help
show options
start over?
accept / reject remaining
accept / reject remaining
accept / reject remaining
accept / reject remaining
accept / reject remaining
Detail

Intake unique	Intake similar	Identical	Hospital similar	Hospital unique
	Acetaminophen PO q6h 32 mg		Acetaminophen PO q4h 325 mg	
		Darbepoetin SC qFriday 60 mg		
		Calcitrol PO daily 0.25 mg		
		Ramipril PO daily 5 mg		
Meloxicam PO daily 7.5 mg				
	Folvite PO daily 1 mg		Folic acid PO daily 1 mg	
				Omeprazole PO daily 40 mg
				Ciproflaxocin PO daily 500 mg
				Ferrous Gloconate PO TID 300 mg

Nothing to display.

Figure 4. Similar items align. Golden-yellow emphasizes the parts that differ, popping a bit from the gray and white background. Notice that only similar items share the same row.

Note: Highlighting helps. In "real" datasets, some attributes may be missing and a block of a different color or a "??" may be used instead to reveal what is missing; use a consistent set of highlighting rules for similar items. Choose an appropriate color: we chose a softer golden-yellow to contrast with our greens, whites, and grays.

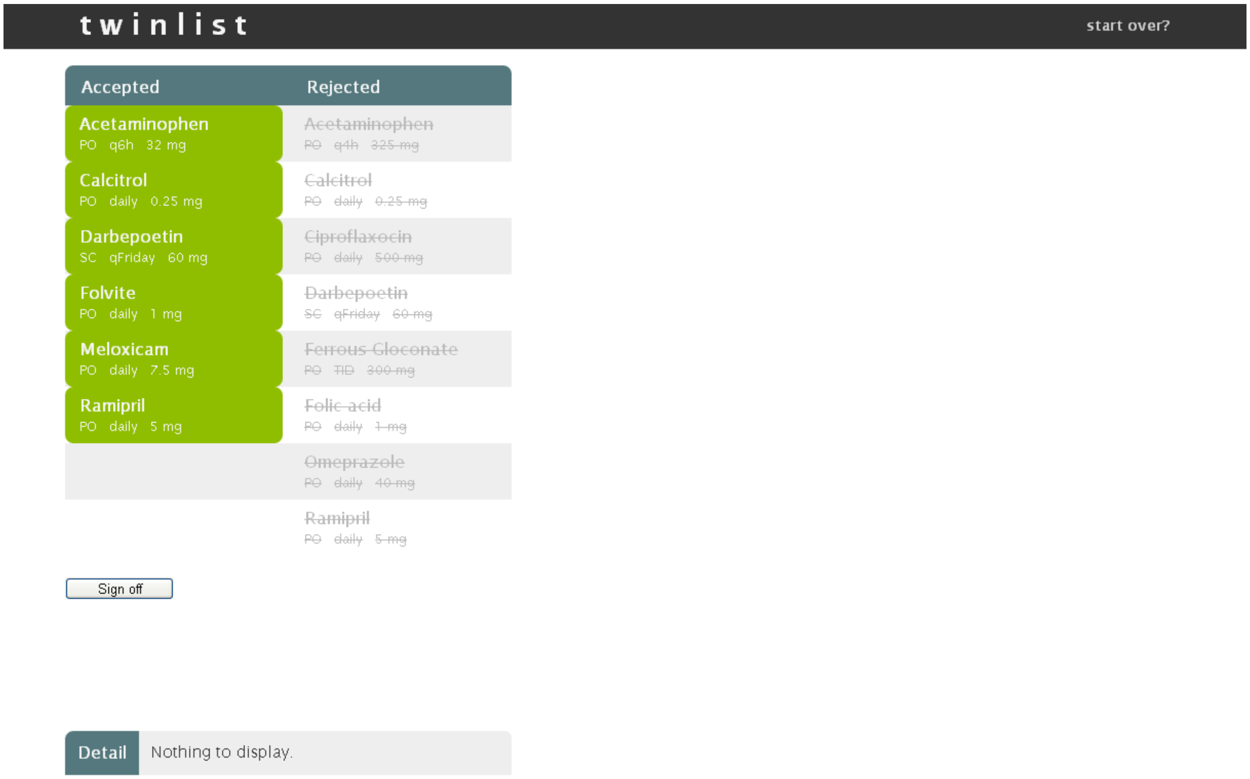

twinlist
compare lists
confirm choices
show help
show options
start over?
accept / reject remaining
accept / reject remaining
accept / reject remaining
accept / reject remaining
accept / reject remaining
Detail

Intake unique	Intake similar	Identical	Hospital similar	Hospital unique
Meloxicam PO daily 7.5 mg		Darbepoetin SC qFriday 60 mg		Omeprazole PO daily 40 mg
		Calcitrol PO daily 0.25 mg		Ciprofloxacin PO daily 500 mg
		Ramipril PO daily 5 mg		Ferrous Gloconate PO TID 300 mg
	Acetaminophen PO q6h 32 mg		Acetaminophen PO q4h 325 mg	
	Folvite PO daily 1 mg		Folic acid PO daily 1 mg	

Nothing to display.

Figure 5. Compact to save vertical space. All identical and unique items are pushed to the top; similar items group at the bottom.

Note: This step does not always help. If all items are either "identical" or "similar", for instance, there is nothing to compact.

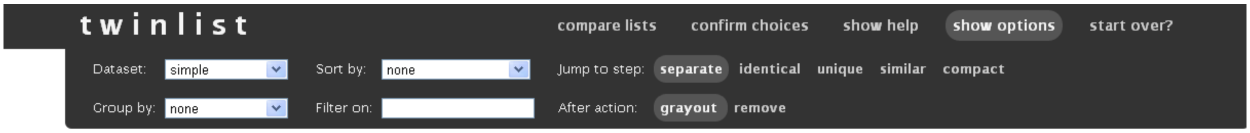


Accepted	Rejected
Acetaminophen PO q6h 32 mg	~~Acetaminophen~~ ~~PO q4h 325 mg~~
Calcitrol PO daily 0.25 mg	~~Calcitrol~~ ~~PO daily 0.25 mg~~
Darbepoetin SC qFriday 60 mg	~~Ciprofloxacin~~ ~~PO daily 500 mg~~
Folvite PO daily 1 mg	~~Darbepoetin~~ ~~SC qFriday 60 mg~~
Meloxicam PO daily 7.5 mg	~~Ferrous Gloconate~~ ~~PO TID 300 mg~~
Ramipril PO daily 5 mg	~~Folic acid~~ ~~PO daily 1 mg~~
	~~Omeprazole~~ ~~PO daily 40 mg~~
	~~Ramipril~~ ~~PO daily 5 mg~~

Detail Nothing to display.

Figure 6. Summary and sign off. A simple summary of the user's decisions to provide a sense of closure. Accepted and rejected styles are consistent with those used in the main interface.

Note: As mentioned earlier, this should be part of the "outer" interface. The above is an adequate overview summary for the physician to sign-off, but patients will need a different summary -- e.g., a layout with three columns: (1) drugs to stop taking at home, (2) drugs to continue, and (3) drugs to start / have been modified.



Dataset: simple

Sort by: none

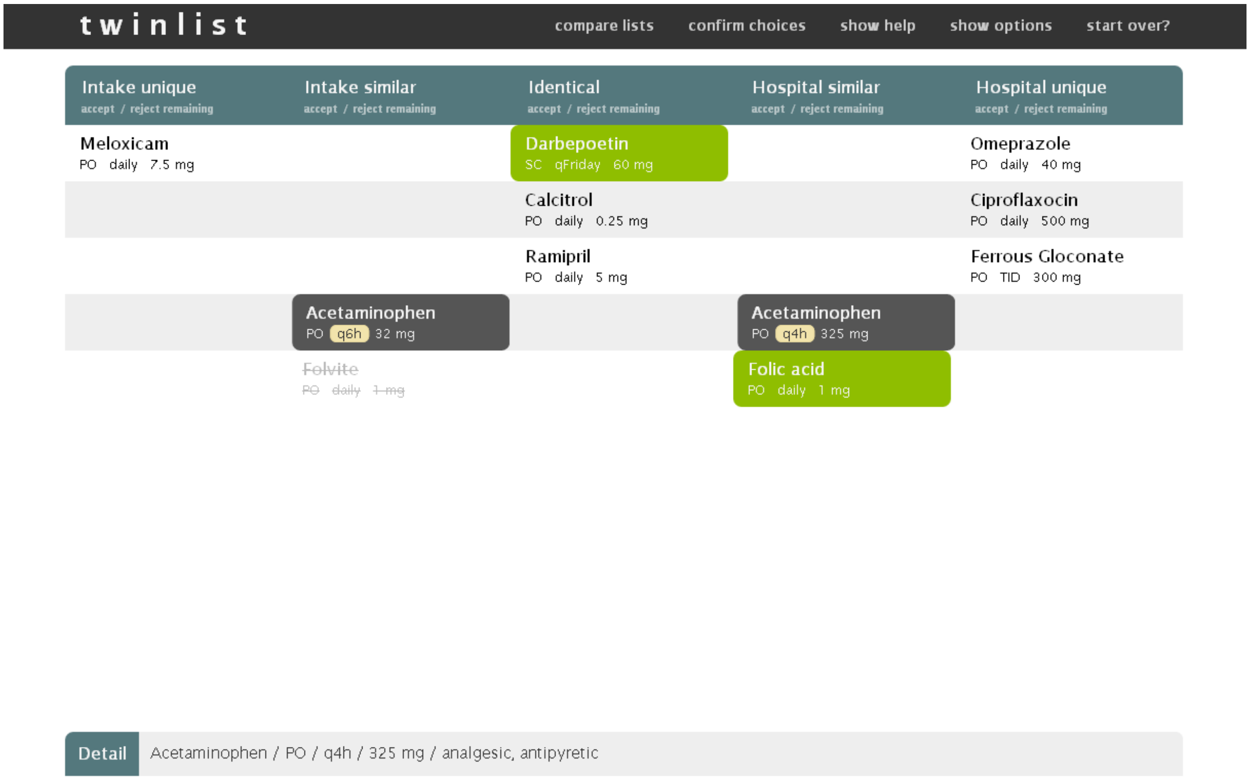
Jump to step: separate identical unique similar compact

Group by: none

Filter on:

After action: grayout remove

Figure 7. Options panel. The options panel, when open, is anchored to the page header, and is styled the same way. The panel's left margin aligns with the left margin observed by the rest of page.



Intake unique accept / reject remaining	Intake similar accept / reject remaining	Identical accept / reject remaining	Hospital similar accept / reject remaining	Hospital unique accept / reject remaining
Meloxicam PO daily 7.5 mg		Darbepoetin SC qFriday 60 mg		Omeprazole PO daily 40 mg
		Calcitrol PO daily 0.25 mg		Ciprofloxacin PO daily 500 mg
		Ramipril PO daily 5 mg		Ferrous Gloconate PO TID 300 mg
	Acetaminophen PO q6h 32 mg		Acetaminophen PO q4h 325 mg	
	~~Folvite~~ ~~PO daily 1 mg~~		Folic acid PO daily 1 mg	

Detail Acetaminophen / PO / q4h / 325 mg / analgesic, antipyretic

Figure 8. Color coding. Yellow-green items are accepted, crossed-out items are rejected, and all other items are undecided. The warm gray item in the center is the one the user is currently hovering over, nudged slightly to the right. Notice how solid-colored blocks come to the forefront while light gray items recede. Such visual coding should account for users with color vision deficiency; double encoding is often useful (above, we indicate rejected drugs with a different color *and* with a strike through the drug name).

Additional note on color schemes: It is often helpful to use colors that users are already familiar with, or colors that are naturally associated with your desired dataset(s). For example our blue-green scheme began with the thought of green scrubs. Dark gray gave the page header some needed weight, and the rest of the color palette evolved from there. Color coding should also account for users with color vision deficiencies; double encoding is often usefu

8 Implementation

Twinlist is written in HTML5, CSS3, and JavaScript (using jQuery): HTML5 for content, CSS3 for polish, and JavaScript for everything else. All datasets are currently hard-coded for demo performance and portability, though it would be trivial to fetch them from an external source if necessary.

Note: The current version performs best in Google Chrome 14+ and runs nearly as well in Mozilla Firefox 5+ and Internet Explorer 9+. Internet Explorer 8 and earlier are not supported. However, the entire application can be downgraded to HTML4 and CSS2 should browser compatibility be at a premium. Contact us if you are interested in *our exploration of a downgraded version for IE8.*

The codebase (excluding .min.js libraries) consists of roughly 3,000 lines across six files:

- `twinlist.html is the base of the application;`
- `5col.css controls appearance and mouseover animations;`
- `init.js bootstraps initialization;`
- `utils.js handles client-side storage (either web storage or cookies, depending on the browser);`
- `model.js encapsulates all view-independent data (including sort order, group order, filter exclusion rules, and so on);`
- `controller.js manages all view-dependent behavior, coordinating all responses to user action (notably redraw()).`

9 Animation

Each animation takes 800 ms, follows an ease-in, ease-out trajectory, and is punctuated with a 600 ms pause. The base duration deviates slightly from the 1 second norm [Robertson et al. 2002], shortened to accommodate the 600 ms necessary to properly separate different stages. The ease-in, ease-out pacing -- based on traditional principles of animation [Lasseter 1987] -- provides aesthetic fluidity and eases object tracking [Dragicevic et al. 2011]. We use a simple cubic easing function.

Note: Larger screen resolutions may stray slightly above 1 second for smoother interpolation; smaller screens could accommodate a somewhat shorter duration, since there is less distance to cover.

Per-item animations are dataset-dependent and necessarily custom-tailored.
The current solution opts for precise control at the cost of added complexity, individually positioning free-moving boxes on top of an empty table. A set of “key frames”, expressed as a series of row and column offsets, encode when and where list items move. These key frames are calculated once on page load, and are recalculated only as needed (e.g., when the window is resized).

Note: An earlier version animated table cells directly, but suffered from unintuitive hover behavior: cells that had moved to new locations still raised mouseover and mouseout events when users hovered over their original, HTML-determined positions.

10 Scalability concerns

Because the number of drugs in the list will never exceed a few dozen items, performance is not likely to ever be an issue. The animation speed may need to be increased in the 1st step where pairs of items are moved one by one, however. One way is to move the 1st pairs slowly and then speed up for the remaining pairs.

With large datasets, all the items may not fit on the screen, requiring scrolling. One way to address the issue is to dynamically reduce the font size so that everything fits (but only up to a point so that readability is maintained). If this is insufficient and scrolling is required, it is critical to make sure users notice the need to scroll with a clear enough scroll bar, possibly supplemented with a visible marker at the bottom of the page indicating there is more below it. The fact that the application checks that all drugs have been acted upon before sign-off is helpful.

The most important thing to handle larger lists gracefully is to detect that similar items that should be highlighted are off-screen. A simple, scalable solution is to simply indicate that some items have fallen out of view, prompting users to scroll to see them. This could be accomplished with an indicator at the bottom or top of each column where more similar items are hidden from view (e.g. an arrow, or a jagged edge, possibly annotated with the number of items).

Note: We are currently exploring possible implementation of those ideas. Stay tuned.

11 Future work and conclusions

We hope to work with vendors to get one implementation of Twinlist working within an operational EHR system in the spring of 2012. This will allow us to conduct some user testing with medical professionals and with real data. We hope to learn how well the animation is received and what kind of options are needed to control it.

We will also add code to handle JSON data streams to facilitate connection to commercial systems. We might also investigate how Twinlist can be applied to other list comparisons, such as allergy or problem lists.

Generalizing the interface to support new domains will raise different issues. Comparing weekly grocery trips, for instance, would benefit from a stronger sense of time: am I paying more or less than before? Is there a general upward or downward trend? Equivalence classes are no longer equally important: unique items could indicate habits to curb (e.g., spontaneous snack purchases), leading to swift and painless budget cuts; similar and identical items could correspond to weekly staples, requiring more nuanced evaluation.

Of particular note is the ability to support significantly longer lists. The current implementation, on a 1024 x 768 pixel screen, supports lists of approximately 10 items. Any more, and the list viewer will scroll to accommodate. Larger screens help. So does the “remove after action” option. But screen space will always remain the primary limiting factor.

Perhaps it would be more interesting to completely re-imagine the interface. The problem naturally lends itself to a variety of interpretations, each with their own respective strengths. One version, for example, could indicate relationships with smooth, curved lines, gently guiding the eye from one medication to another. This version would automatically indicate when relevant medications fall off-screen, and could support a variety of list representations (e.g.

two horizontal lists, one combined list, even a radial layout). But the lines also introduce visual clutter and limit the number of relationship types that can be effectively encoded, particularly for densely interrelated lists.

11 Availability

The demo and code is available on request. Please contact Catherine Plaisant plaisant@cs.umd.edu

The video presented at the American Medical Informatics Association (AMIA) 2011 Annual Symposium, is available at:
http://www.youtube.com/watch?v=YoSxIKl0pCo&noredirect=1

and can be downloaded at:
http://www.cs.umd.edu/hcil/sharp/twinlist/twinlistamia7.mp4

The code is available under the MIT license (see http://en.wikipedia.org/wiki/BSD_licenses). jQuery also uses the MIT license.

The full copyright text that we use is as follows:

```
Copyright (c) 2011 Human-Computer Interaction Lab, University of Maryland, Tiffany Chao,
Catherine Plaisant and Ben Shneiderman, http://www.cs.umd.edu/hcil/sharp

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

See also the MIT license statement for jQuery**:**

```
Copyright (c) 2011 John Resig, http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
```

12 Acknowledgements

We would like to thank Leonardo Claudino, Sameh Khamis, Ran Liu, Ben London, and Jay Pujara for developing an early version of Twinlist. Their work, completed for the Spring 2011 Information Visualization course, can be found on their wiki. A copy of their Workshop on Interactive Systems in Healthcare (WISH) 2011 paper is available on the HCIL project website.

This work is supported in part by Grant No. 10510592 for Patient-Centered Cognitive Support under the Strategic Health IT Advanced Research Projects Program (SHARP) from the Office of the National Coordinator for Health Information Technology.