Visual techniques for medical reconciliation: spatial metaphor, animated explanation, and flexible decision-making

Tiffany Chao, tchao@cs.umd.edu (November 21, 2011)
Department of Computer Science Honors Paper
University of Maryland, College Park, MD

Abstract

Medication reconciliation can help prevent adverse health outcomes, but the process is nontrivial. At its core is a list reconciliation problem: take *n* lists, review all *j* items, and decide which *k* items to keep. That alone, however, can be mentally taxing and error-prone -- especially so in the medical context, in which discrepancies of any kind may be clinically significant. Twinlist offers a fresh perspective on what an effective medication reconciliation interface could be like, leveraging animation and spatial metaphor to visually clarify, and thereby simplify, the core reconciliation problem.

1 Introduction

There are three kinds of medication errors:

1. harmful (*preventable adverse drug events*, or *PADEs*),
2. potentially harmful (*near-misses*, either intercepted or avoided by sheer luck), and
3. harmless (the most common) [Keohane and Bates 2008].

By definition, only (1) results in harm; unfortunately, at least 1.5 million of them occur every year [CIPME 2007].[1]

Patients are particularly vulnerable to errors at care transitions [Rozich and Resar 2001, Forster et al. 2003], where medication regimens frequently change. Properly reconciling medications at these points is crucial. But complete and accurate reconciliation is markedly difficult, and thus often overlooked or simply not performed [Rogers et al. 2006].

[1] Specific figures vary widely depending on error detection methodologies, are based on different study definitions, and are typically cited according to disparate metrics (e.g. errors per order, errors per 1,000 patient days, or errors per 1,000 patient admissions). The hospital setting is the best researched by far: landmark studies include Leape et al. 1991, Bates et al. 1995, Classen et al. 1997, and Jha et al. 1998.

The plain truth is that there are no best practices -- only isolated instances of success.[2] Few papers propose actual interfaces; fewer still address the act of reconciliation itself. This paper aims to do both.

2 The reconciliation problem

Medication reconciliation, in the general sense, consists of three parts:

1. verifying medication use,
2. identifying any discrepancies, and
3. rectifying any resultant errors [Vira et al. 2006].

The "reconciliation" that this paper addresses, however, is purely the act of comparing one medication regimen to another -- a combination of (2) and (3).[3]

This reconciliation is not easy. The same medication may have been prescribed on different dates, under different names, with different dosages, for different reasons; conversely, completely separate medications, lacking any obvious similarities, may have been prescribed to treat the same condition. Every variance matters. And it is the caregiver's responsibility to ensure that they are properly resolved -- all while never forgetting the intended therapeutic plan.

Poorly designed interfaces only add to the cognitive burden. Many commercial systems are highly impractical or outdated, riddled with convoluted menus and fragmented presentations of data. Most of them are simply ill-equipped to support clinician workflows, instead opening new paths to adverse health outcomes [CIPME 2007].

3 Related work

The closest example in the medical literature is the Pre-Admission Medication List (PAML) Builder [Poon et al. 2006]. This interface, like contemporary reconciliation interfaces, presents all medications from all sources in one combined "superlist", grouped alphabetically by generic name. And like contemporary reconciliation interfaces, it exhibits a visual homogeneity that does very little to help clinicians identify where variances could be.[4]

---

[2] And there are many. See, for instance, Rozich and Resar 2001, Pronovost 2003, or Vira et al. 2006. For an overview, refer to CIPME 2007.

[3] Much of the literature focuses on (1): quickly and accurately assimilating information from multiple sources of variable integrity is no small feat. Recent efforts to reduce history-taking errors include Lesselroth et al. 2009, Gizzi et al. 2010, and De Winter et al. 2011.

[4] For more information, see Markowitz et al. 2011.

Representations of relatedness from other domains are more successful. The Semantic Graph Visualizer [Andrews et al. 2009] encodes structural similarities and differences with color, juxtaposing source and reconciled graphs for natural pairwise comparisons. Jigsaw [Stasko et al. 2008] places special emphasis on visualizing connections, coordinating various views to encourage broad, dynamic document exploration. Parallel Tag Clouds [Collins et al. 2009] use faint, weighted "stubs" to indicate related items, converting these "stubs" to complete lines only when the user focuses on a particular list item; sizing by rank or score also intuitively assigns visual salience based on relative importance.

Animation is less straightforward. Done well, it is compelling, expressive, entertaining; directing attention, promoting object constancy, and making even complex transitions intelligible.[5] Done poorly, it distracts, confounds, irritates; wastes time, reduces accuracy, and is nothing but disappointing.[6]

Visualization-safe animations are hard to do right. But Twinlist is willing to bet on their superlative storytelling potential.

4 The interface

Twinlist's interface[7] 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.

Interaction begins with an introductory window that swipes down from the top of the screen. Once the user closes the window, the two lists to be reconciled sweep in from the top left corner and settle into two distinct columns. The leftmost column, **Intake**, lists what the patient was taking before he was hospitalized; the rightmost column, **Hospital**, describes what the patient took during his stay.

---

[5] Animation seems particularly good in this last capacity: Robertson et al. 1991, Yee et al. 2001, Robertson et al. 2002, and Heer and Robertson 2007 (among others) corroborate this trend. Not that animation is only good for transitions. Bederson and Boltman 1999 demonstrate that it can help users understand and reconstruct data spaces, for example; Forlizzi et al. 2003 use its expressive capacity to support affective, text-based communication.

[6] Tversky et al. 2002 is one of various studies that cast doubts on animation's usefulness. Fisher 2010 summarizes both viewpoints.

[7] A video of the latest version, presented at the American Medical Informatics Association (AMIA) 2011 Annual Symposium, is available on YouTube.

Clicking **compare lists**, as instructed in the introductory window, triggers the list reconciliation animation (Figures 1 - 5).[8] Once the animation completes, users may click to accept or reject medications as they see fit. When they 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.

The most important operations -- accept and reject -- are available with mouse clicks: left-click to accept, right-click to reject, and double-click to undo.[9] Users may left- or right-click as they please, and are free to undo or redo as many times as they wish. Explicit **accept** and **reject** buttons beneath column headers provide a convenient way to accept or reject entire columns as appropriate.[10]

Advanced features are hidden by default, accessible only via **show options** (Figure 7). Here users may customize the list viewer to suit their needs: they may sort by name or dosage, group by route or therapeutic intent,[11] and/or filter on any fragment of a medication's name; replay any stage of the list reconciliation animation; and choose to remove all accepted and rejected medications to save even more screen space. As before, users may apply or remove any combination of options at any time.

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.

---

[8] 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].

[9] Clicking on a medication toggles through three decision 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.

[10] It is unclear whether column-accept and column-reject should overwrite previous decisions; the current implementation errs on the side of caution and only applies the command to "undecided" medications.

[11] Currently, medications only belong to one such group. Supporting *n*-group affiliation would require the medication to be appear *n* times, once for every group.

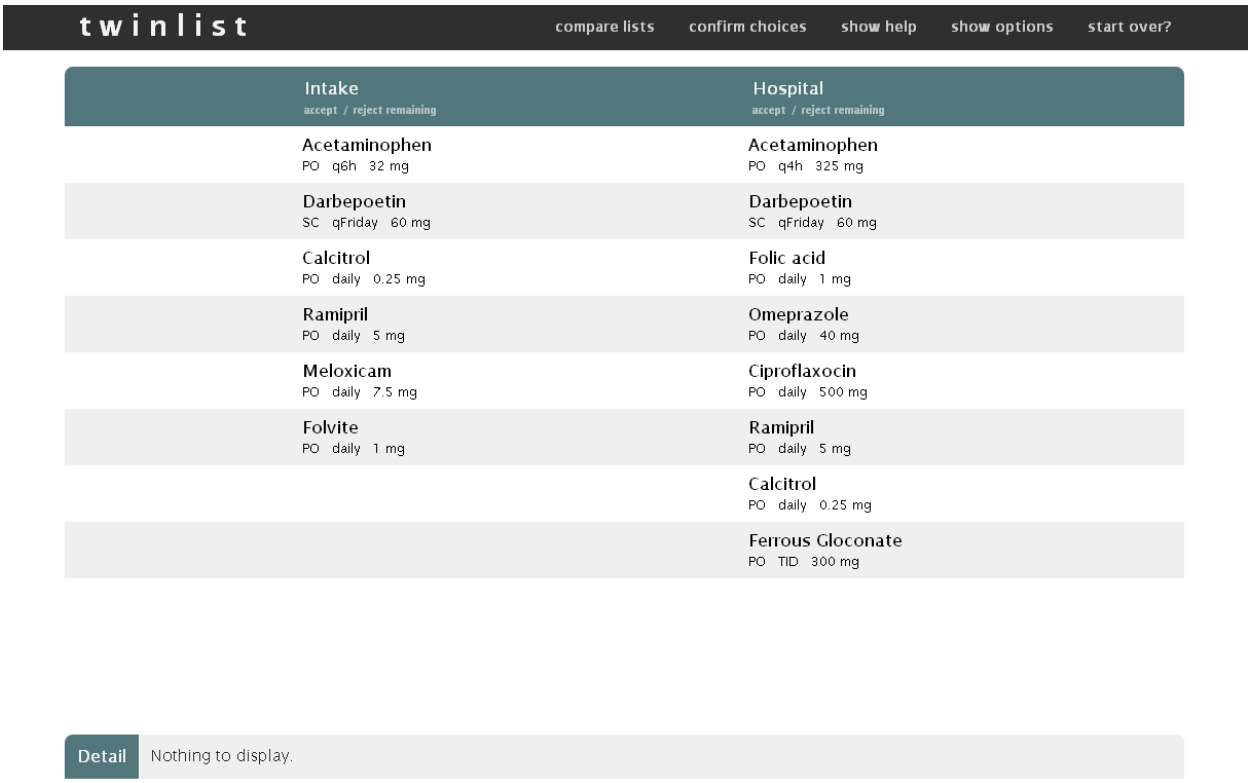

Figure 1. Separate lists. The two lists start as two separate columns, **Intake** and **Hospital**. List items are unsorted by default.

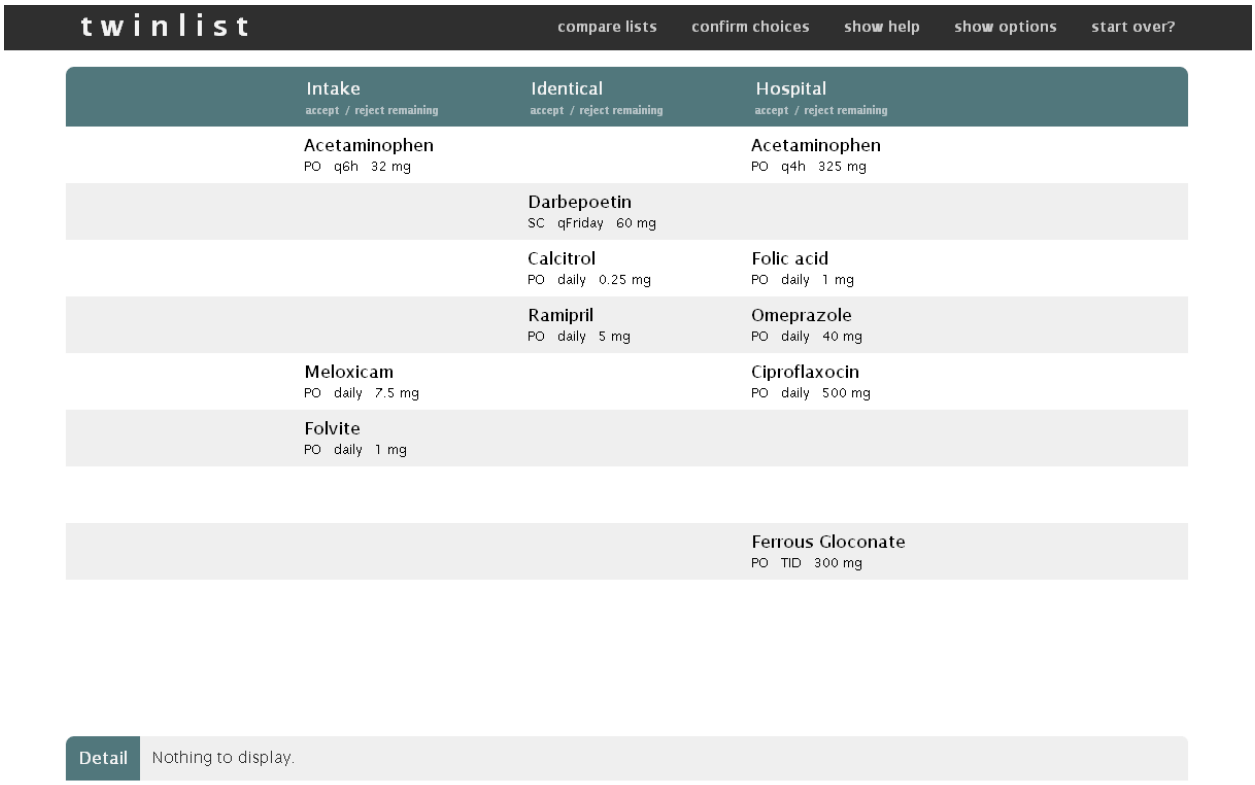

Figure 2. Identical items converge. Items from the same set animate in unison; the sets themselves animate sequentially. Each set waits for its turn: items from the first set fly to the center and converge, then items from the second set fly to the center and converge, and so on. Note that the position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

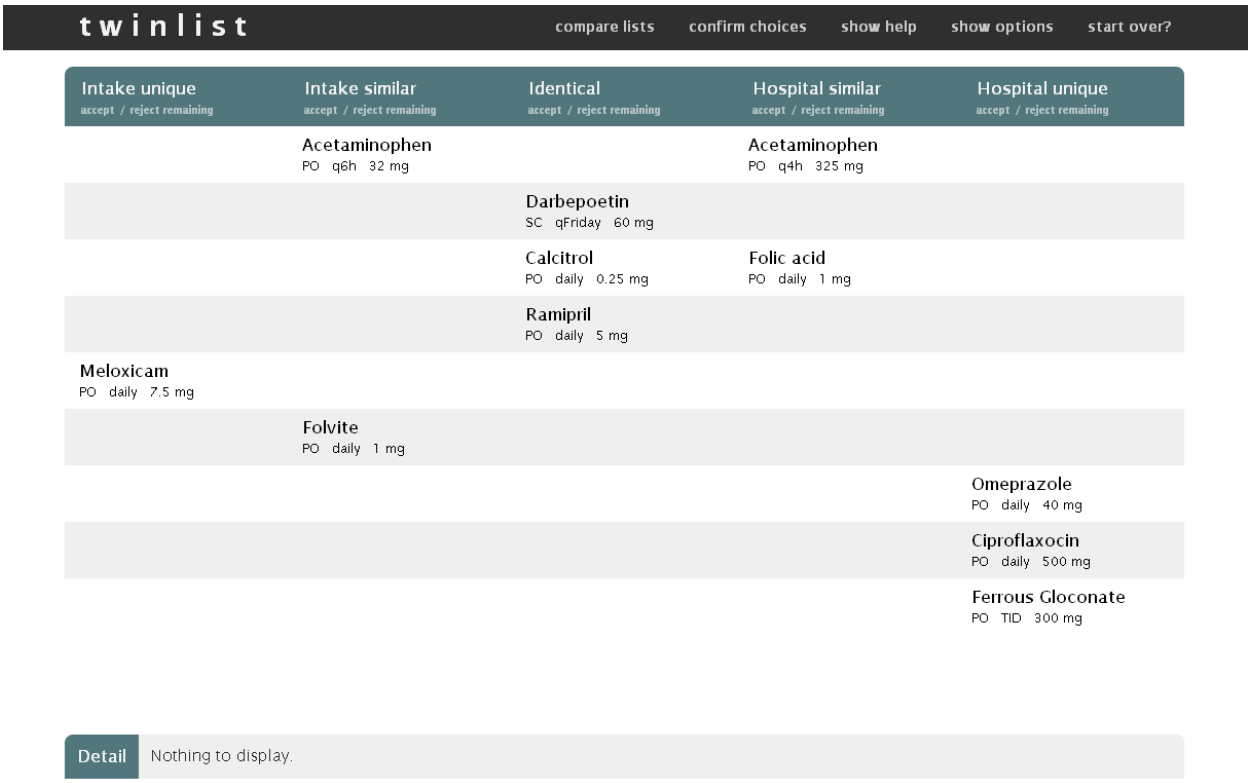

Figure 3. Separate unique items. Medications that are unique to either list fly to the leftmost and rightmost columns. This transition is a two-part animation: first all items that are unique to **Intake** fly to the left, then all items that are unique to **Hospital** fly to the right.

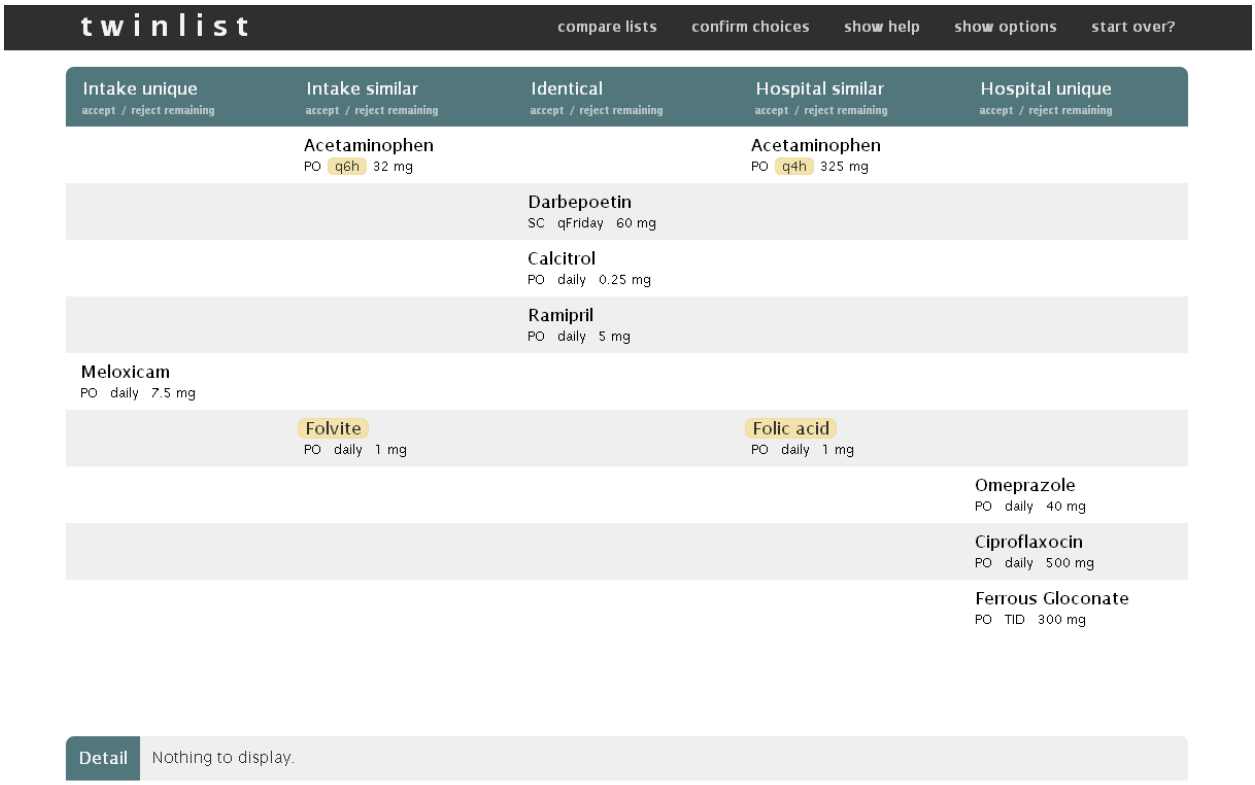

Figure 4. Similar items align. Golden-yellow emphasizes the parts that differ. Notice that only similar items share the same row.

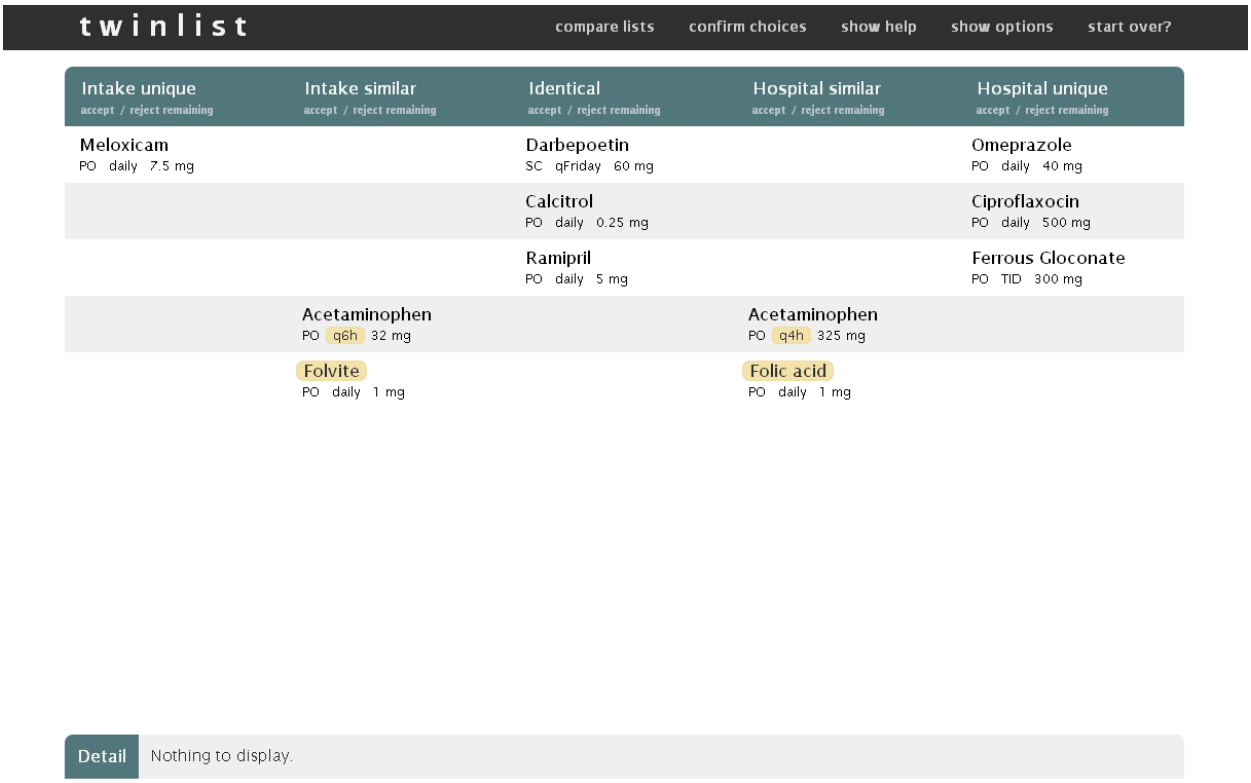

Figure 5. Compact to save vertical space. All identical and unique items are pushed to the top; similar items group at the bottom.

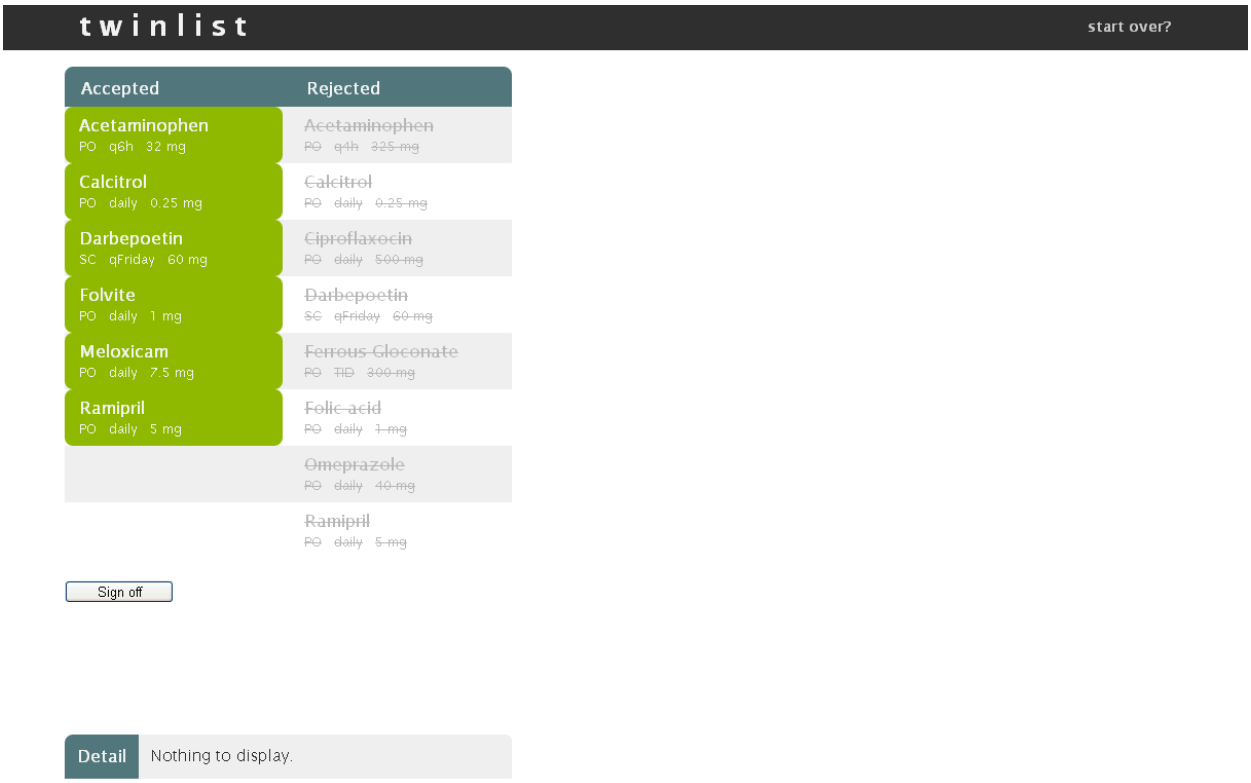

Figure 6. Summary and sign off. A simple summary of the user's decisions provides a sense of closure. Accepted and rejected styles are consistent with those used in primary interaction (see Figure 8).

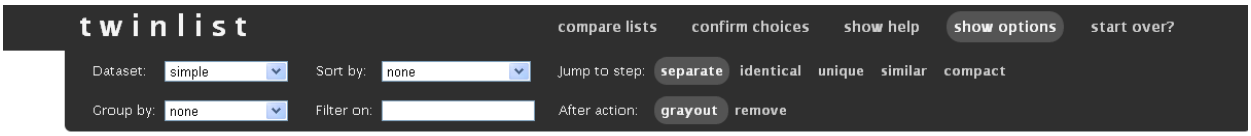

Figure 7. Options panel. The options panel, when open, is anchored to the page header. The panel's left margin aligns with the left margin observed by the rest of page.

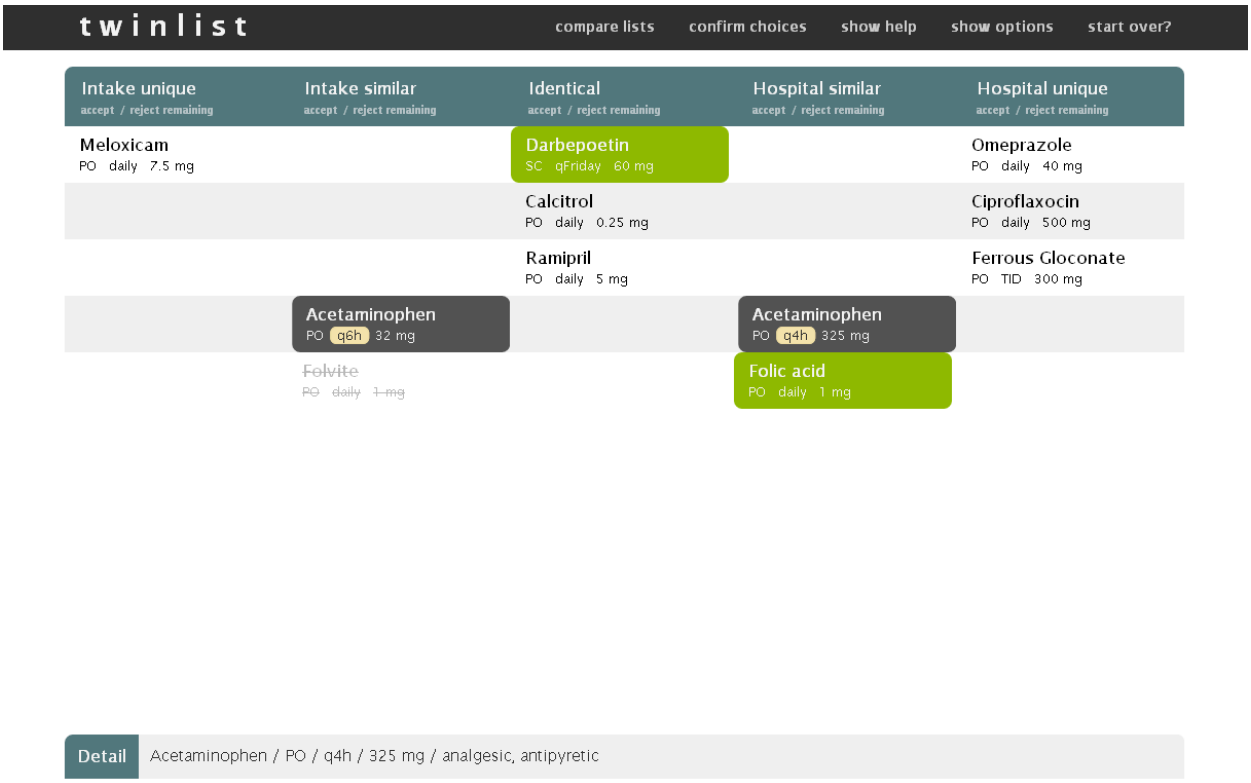

Figure 8. Color coding. Yellow-green items are accepted, crossed-out items are rejected, and all other items are undecided. Warm gray items, nudged slightly to the right, are related to the medication that the user is currently hovering over. Notice how solid-colored blocks come to the forefront while light gray items recede.

5 Implementation

Twinlist is written in HTML5, CSS3, and JavaScript (using jQuery): HTML5 for content, CSS3 for polish, and JavaScript for everything else.[12] 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.

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;

[12] 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. The entire application could be downgraded to HTML4 and CSS2, however, should browser compatibility be at a premium.

- `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()`).

Medications are functionally defined as a set of five attributes (`name`, `route`, `frequency`, `dosage`, and `purpose`), and are each assigned a unique numeric `id`. List items are thus `similar` if between one and four attributes match, `identical` if all five attributes are exactly the same, and `unique` if they are neither similar nor identical to any medication in either list. Note that similar or identical items must also share at least one active ingredient.[13]

Per-item animations are dataset-dependent and necessarily custom-tailored. The current solution[14] 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;[15] these key frames are calculated once on page load, and are recalculated only as needed (e.g., when the window resizes).

6 Conclusion

Detailed user feedback is forthcoming. But the long process of designing and tweaking the interface can be summarized in the following recommendations:

1. *Keep it (visually) simple.*

   Seriously consider whether the animation is worth the complexity. Then evaluate whether it is worth a steep time penalty. Then judiciously allocate visual salience. In particular, avoid unnecessary motion.

2. *Do one thing at a time.*

[13] Data providers must explicitly specify this information, as Twinlist is deliberately algorithm independent.

[14] The previous 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.

[15] The conversion from (row, col) offsets to precise (x, y) pixel coordinates is as follows:

```
pixels from left ← col offset * width of background table cell
pixels from top  ← row offset * height of background table cell + header offset
```

The above is only calculated when an animation is set to play.

Break complex transitions into atomic segments. Allocate approximately 1 second per transition; reserve another 750 ms to properly punctuate separate stages.

3. *Style purposefully.*

   Design can either break or make the interface. Line, color, texture, form, space -- all of these can make (1) easier to achieve. Meaningless colors and shapes are wasted opportunities.

4. *Stage everything.*

   Multi-part animations benefit greatly 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 a natural sequence of events. Carefully direct user attention to lessen the burden of comprehension.

7 Future work

The obvious next step would be to evaluate the system -- get more feedback, drive new ideas. Input from those that would stand to benefit would be ideal.

Preparing for commercial adoption would introduce a host of new practical considerations. It would need to integrate with existing medical systems,[16] connect to various databases, accept incomplete or malformed data; accommodate a broader spectrum of platforms, screen resolutions, and clinician populations; allow users to modify, look up, and manually reshuffle medications; and ground itself with patient information. It could be extended to let physicians reconcile any number of lists in pairs of two, visually emphasize a “criterion standard”, incorporate a dimension of dataset confidence -- the list goes on.

Generalizing the interface to support new domains raises 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.

[16] In particular, electronic health records and computerized physician order entry systems.

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.

Practicality side, however, there are many other questions to consider. Is the layout more useful than the animation, for instance? Is the animation only helpful when users are learning the interface? How does the current strategy compare to other representations of similarity and dissimilarity? How could the interface be extended to support general list reconciliation? These, and other questions, pose interesting problems for future study.

Acknowledgements

The author would like to thank Drs. Atif Memon, Catherine Plaisant, Ben Shneiderman, and Michelle Hugue for their continued support. Thanks also go to Leonardo Claudino, Sameh Khamis, Ran Liu, Ben London, and Jay Pujara for developing the original prototype.[17] 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.

References

[1] K. Andrews, M. Wohlfahrt, and G. Wurzinger, “Visual Graph Comparison,” in *Proceedings of the 2009 IEEE Symposium on Information Visualization*, 2009, pp. 62-67.

[2] D. W. Bates et al., “Incidence of adverse drug events and potential adverse drug events. Implications for prevention. ADE Prevention Study Group.,” *The Journal of the American Medical Association*, vol. 274, no. 1, pp. 29-34, Jul. 1995.

[3] B. B. Bederson and A. Boltman, “Does animation help users build mental maps of spatial information?,” in *Proceedings of the 1999 IEEE Symposium on Information Visualization*, 1999, vol. 11, no. 98, pp. 28-35.

[17] 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 University of Maryland's Human-Computer Interaction Lab website.

[4] D. C. Classen, S. L. Pestotnik, R. S. Evans, J. F. Lloyd, and J. P. Burke, “Adverse drug events in hospitalized patients,” *The Journal of the American Medical Association*, vol. 277, no. 4, pp. 301-6, May 1997.

[5] L. Claudino, S. Khamis, R. Liu, B. London, and J. Pujara, “Facilitating Medication Reconciliation with Animation and Spatial Layout,” *hcil.cs.umd.edu*, pp. 2-6, 2011.

[6] C. Collins, F. B. Viegas, and M. Wattenberg, “Parallel Tag Clouds to explore and analyze faceted text corpora,” in *Proceedings of the 2009 IEEE Symposium on Visual Analytics Science and Technology*, 2009, pp. 91-98.

[7] Committee on Identifying and Preventing Medication Errors (CIPME), *Preventing medication errors*. Washington, DC: National Academies Press, 2007.

[8] S. De Winter et al., “A simple tool to improve medication reconciliation at the emergency department,” *European Journal of Internal Medicine*, vol. 22, no. 4, pp. 382-5.

[9] P. Dragicevic, A. Bezerianos, W. Javed, N. Elmqvist, and J. D. Fekete, “Temporal distortion for animated transitions,” in *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, 2011, pp. 2009–2018.

[10] D. Fisher, “Animation for Visualization: Opportunities and Drawbacks,” in *Beautiful Visualization: Looking at Data Through the Eyes of Experts*, O’Reilly Media, Inc., 2010, pp. 329-352.

[11] J. Forlizzi, J. Lee, and S. Hudson, “The kinedit system,” in *Proceedings of the Conference on Human Factors in Computing Systems*, 2003, no. 5, p. 377.

[12] A. J. Forster, H. J. Murff, J. F. Peterson, T. K. Gandhi, and D. W. Bates, “The incidence and severity of adverse events affecting patients after discharge from the hospital,” *Annals of Internal Medicine*, vol. 138, no. 3, pp. 161-7, Feb. 2003.

[13] J. Heer and G. Robertson, “Animated transitions in statistical data graphics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240-7, 2007.

[14] A. K. Jha et al., “Identifying adverse drug events: development of a computer-based monitor and comparison with chart review and stimulated voluntary report,” *Journal of the American Medical Informatics Association*, vol. 5, no. 3, pp. 305-14, 1998.

[15] C. A. Keohane and D. W. Bates, “Medication safety,” *Obstetrics and Gynecology Clinics of North America*, vol. 35, no. 1, pp. 37-52, viii, Mar. 2008.

[16] J. Lasseter, “Principles of traditional animation applied to 3D computer animation,” *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 35-44, Aug. 1987.

[17] L. L. Leape et al., “The nature of adverse events in hospitalized patients,” *The New England journal of medicine*, vol. 324, no. 6, pp. 377-84, Feb. 1991.

[18] B. J. Lesselroth et al., “Design and implementation of a medication reconciliation kiosk: the Automated Patient History Intake Device (APHID),” *Journal of the American Medical Informatics Association*, vol. 16, no. 3, pp. 300-4, 2009.

[19] Markowitz, E., Bernstam, E., Herskovic, J., Zhang, J., Shneiderman, B., Plaisant, C., Johnson, T., “Medication Reconciliation: Work Domain Ontology, Prototype Development, and a Predictive Model,” *Proc. AMIA Annual Symposium 2011*, pp. 878-887.

[20] E. G. Poon et al., “Design and implementation of an application and associated services to support interdisciplinary medication reconciliation efforts at an integrated healthcare

delivery network," *Journal of the American Medical Informatics Association*, vol. 13, no. 6, pp. 581–592, 2006.

[21] P. Pronovost, "Medication reconciliation: a practical tool to reduce the risk of medication errors," *Journal of Critical Care*, vol. 18, no. 4, pp. 201-205, Dec. 2003.

[22] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone trees: animated 3D visualizations of hierarchical information," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1991, vol. 189, p. 194.

[23] G. Robertson, K. Cameron, M. Czerwinski, and D. Robbins, "Polyarchy visualization: visualizing multiple intersecting hierarchies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2002, no. 1, pp. 423–430.

[24] G. Rogers et al., "Reconciling medications at admission: safe practice recommendations and implementation strategies," *Joint Commission Journal on Quality and Patient Safety / Joint Commission Resources*, vol. 32, no. 1, pp. 37-50, Jan. 2006.

[25] J. Rozich and R. Resar, "Medication safety: one organization's approach to the challenge," *Journal of Clinical Outcomes Management*, vol. 8, no. 10, pp. 27–34, 2001.

[26] J. Stasko, C. Görg, and R. Spence, "Jigsaw: supporting investigative analysis through interactive visualization," *Information Visualization*, vol. 7, no. 2, pp. 118-132, 2008.

[27] B. Tversky, J. B. Morrison, and M. Betrancourt, "Animation: can it facilitate?," *International Journal of Human-Computer Studies*, vol. 57, no. 4, pp. 247–262, 2002.

[28] T. Vira, M. Colquhoun, and E. Etchells, "Reconcilable differences: correcting medication errors at hospital admission and discharge," *Quality & Safety in Health Care*, vol. 15, no. 2, pp. 122-6, Apr. 2006.

[29] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst, "Animated exploration of dynamic graphs with radial layout," in *IEEE Symposium on Information Visualization*, 2001, pp. 43-50.