

# Using the Wave Protocol to Represent Individuals' Health Records

Shirley Gaw ([sgaw@google.com](mailto:sgaw@google.com))

Umesh Shankar ([ushankar@google.com](mailto:ushankar@google.com))

There are several challenges in aggregating health records from multiple sources, including merging data, preserving proper attribution, and allowing corrections. Both PHRs and Health Record Trusts must properly audit data provenance so that users, whether health providers to patients, can understand the evolution of health record state and who caused which transitions. Unfortunately, standards for exchanging medical records data, such as CCR and CCD, tend to focus on representing particular clinical data as some subset of a patient's complete record. This provides a snapshot of its state, but there is very little to describe how a sequence of changes to the record should be interpreted as a coherent whole, nor is there a nice way to incorporate corrections or notes from the patient.

Even if individual items in CCR or CCD representations are signed (in fact, institutions generating CCR and CCD often do not), that's not enough to understand the context in which the item was inserted. Was it a new item? Was it an update to an old one? Was there a larger set of items, of which some have been deleted? Simple signatures don't tell us any of this.

CCR and CCD are byproducts of walled gardens of care, where all inputs to the system came from the inside and were made by users authenticating to the same system. They do not match the more dynamic, messy, and distributed nature of aggregation---but this is precisely what end users, patients, actually want. And unfortunately this issue is not solved by the technology sector's currently used protocols, such as REST-style (Google Health) and Database-style (Microsoft HealthVault) APIs in use.

Fortunately, there is something available now that gives us the data aggregation, conflict resolution, and audit trail that what we want: the Google Wave federation protocol. It's built from the ground up to collate multiple sources into a coherent whole.

Here is how the protocol's features address some of our problems:

- *A sequence of updates results in a well-defined final state.* This is independent of how many writers exist or how little the writers communicate with each other, which

fits well with the distributed electronic health record model.

- *Merging of updates and resolving conflicts is handled using Operational Transforms.* Electronic health record implementations do not have to reinvent the wheel: we argue that the merging of health records maps to the collaborative document editing problem---understanding that someone else has created a theoretical framework to handle the details of aggregating the actions of many writers makes data provenance much easier to document and implement in real systems.
- *Ability to "play back" allows for more secure, more usable, attribution.* Patient-controlled records such as PHRs threaten providers: who knows what will be changed by non-educated actors? At the same time, errors in medical records are common, and both providers and patients may need to correct this limitation. For everyone's comprehension of a health record, the ability to correct a record must be accompanied by the ability to replay the sequence of events and understand which actors caused those actions.
- *Federation allows different authoritative providers,* so both tech companies and HMOs can start offering this right away, while preserving integrity by deciding who else to trust.
- *Wave understands XML tags, so existing formats---at a fine enough granularity---can be largely reused.* CCR and CCD need not change; we are discussing how to audit the snapshots and the transitions between them. Electronic health record systems may implement whatever underlying data store they want, but the communication of how to change the "system view" of a record and how it arrived at that state can be handled by the Wave protocol along with existing electronic health record representation formats.

Inevitably, aggregators will get messy or overlapping data, so one of their main challenges is to filter this data and present it to the user in a sensible way. By

using the Wave protocol, any automatic edits to this end made by an aggregator will be visible and well-defined, and they may be undone in a principled way. Naturally, some of this can be done in postprocessing and in the UI, but the exported data should be relatively “clean” as well, and that requires actual edits. Wave allows external participants to understand the full state of the system and therefore only to send new, relevant data; without this, some other complex mechanism would be required to decide what data counts as “new”. In the Wave federation model, attribution is done at the level of providers, which makes sense given that providers are the ones doing authentication; it also eliminates the need for individual users to have their own credentials such as certificates..

While this proposal is meant to be thought-provoking, it is actually quite practical, implementation-wise. The code for the federation server and proof-of-concept Wave server has been open-sourced. Novell has already built a collaboration platform, Novell Pulse, on top of Wave.

This storage model does entail a higher degree of transparency than existing model, since all contributing providers can see your history. Providers can edit data send from other providers, but this is arguably a feature; later visits/second opinions should be able to update earlier diagnoses. One of the main obstacles to transparency, which was patients’ concerns that insurance companies would use the data to exclude them based on pre-existing conditions, has been mitigated by the recent health reform bill.

References:

HL7 CDA v2:

<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1380194/?tool=pubmed>

CCR: <http://www.ccrstandard.com/>

CCD (nearest approximation; it’s not open):

<http://www.hl7.org/Special/committees/structure/index.cfm>

Google Health API:

<http://code.google.com/apis/health/>

Healthvault API:

<http://www.healthvault.com/HealthVaultSDK>

Wave protocol: <http://www.waveprotocol.org/>