ADDIS 2.x requirements and architecture

March 19, 2013

1 Introduction

ADDIS long-term vision Aggregate Data Drug Information System (ADDIS) aims to support transparent, high quality, and easy to use software to support decisions regarding multiple treatment options, such as market authorization and reimbursement decisions. Importantly, results and recommendations should be directly traceable to the underlying evidence (i.e. clinical trials). To this end, ADDIS must capture the entire process in a single integrated workflow. Additional challenges include developing appropriate statistical and decision analytical methods and enabling efficient access to, and management of, the evidence. ADDIS is open source software to encourage third party contributions and enable independent auditing.

Current status A proof-of-concept system (ADDIS 1.16, `http://drugis.org/addis`) has been developed in the context of the TI Pharma project Escher (T6-202). Escher focussed on the market authorisation process and consequently ADDIS enables benefit-risk decisions based on clinical trials. ADDIS implements a database of aggregate-level clinical trial results, network meta-analysis to synthesise the results of multiple trials, and stochastic multi-criteria decision models to support complex benefit-risk decisions.

The proof-of-concept is a cross-platform desktop application developed in Java. This enabled rapid prototyping by limiting concerns of infrastructure, distribution, and security. However, this limits our ability to integrate components not written in Java and makes it difficult to collaborate on a single data set. To increase our flexibility in supporting future research projects and to increase the real-world applicability of ADDIS, we need to transition from the current prototype to a new architecture.

2 Requirements

We identify the main goals and requirements of the 'future system' according to five key concerns: research, valorization, development, ecosystem, and learning.

Research The system should be an enabler for our own research and contribute to the success of grant proposals.

R1 Rapid prototyping of new analyses and user interfaces to support new research questions

R2 A large database to do research and meta-research

R3 Transparency of implementation (open source)

R4 Publishable (high quality) graphics and tables

Valorization The system should generate sufficient revenue to support its continued development and operation. Since the business model has not been decided on, we focus on concerns that are likely to be common to potential approaches.

V1 Custom variants for different audiences, such as regulators, reimbursement authorities and industry (product line)

V2 'In-house' deployments of the system for when the analyses are confidential / sensitive

V3 Access to internal organization data, possibly in conjunction with public data

V4 Ease of use: the system should visually attractive to 'sell' it, and be easy to use to limit frustrations and the need for training

Development System development should be efficient, agile, and sustainable.

D1 Analyses are often computationally intensive, they should run quickly and the system should scale to support many such computations in parallel if needed

D2 There should be flexibility in choosing the right technologies, platforms, and frameworks to implement specific functionality (e.g. statistical computing using R and user interfaces using HTML + JavaScript)

D3 The system should be maintainable and hence divided up in loosely coupled components with well-defined APIs

D4 To ensure our public APIs are functioning in an optimal manner, our own software should use those APIs

D5 The integrity of data should be closely guarded and all data should have clear provenance and versioning information

Ecosystem For the ADDIS concept to work, it is critical that structured clinical trials data are available. It is unlikely that any one organization will be able to achieve this. Therefore, we should aim to 'bootstrap' an open ecosystem in which structured clinical trials data can be shared.

E1 Create a single collaborative portal for data extraction and sharing (open access / source)

E2 Enhance the efficiency of systematic review by 'closing the information chain' (i.e. capturing the entire process from literature screening to analysis)

E3 Enable the flexible re-use of data by enabling the definition, re-definition and mapping of meta-data – it should be possible to integrate data extracted by independent teams by mapping the meta-data

E4 Enable automated (third party) data extraction systems (bots) to contribute

E5 The system(s) should be easy to use and hassle-free to use – e.g. single sign-on

E6 All data and analyses can be traced back to their ultimate source

E7 Third parties should be able to develop new analysis tools and decision support systems based on the available data

Learning The system should promote the use of structured, transparent, and quantitative methods in health care policy decisions.

L1 Enable access to 'complex' methods and tools through a usable interface for non-experts

L2 Clear in-system documentation and links to further reading (e.g. methodology papers)

L3 Raise awareness of newer statistical and decision making methods by implementing and applying them

3 Future architecture

Given that many of the requirements imply loosely coupled and flexibly reusable components with well-defined public APIs (R1, V1, V2, V3, D1, D2, D3, E4, E7), we will assume that a service-based (and web-based) architecture is most appropriate. We distinguish the following components:

- Web services that implement the different analyses (R1, D1, D2)
- ADDIS, a 'business intelligence' system for drug benefit-risk analysis (V1, D3, L1) – basically a 'work flow engine' in which different ways of going from a database of trials to analyses can implemented
- TrialVerse, a portal/database where researchers share structured RCT data (V3, D3, D4, E1)
- TrialMine, a system for literature screening (E2)
- ConceptMapper, a shared component where definitions (concepts) can be deposited, refined, and mapped (R2, E1, E2, E3)
- A user management component underlying all systems (V4, D3, E5)

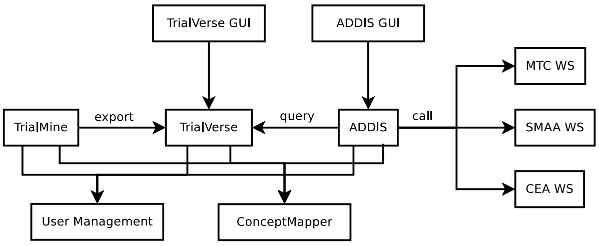

Figure 1: Overview of the possible future architecture. GUI = Graphical User Interface, RCT = Randomized Controlled Trial, WS = Web Service

The user-facing systems have a back end that exposes an API, and an HTML + JavaScript GUI that calls that API (D2, D4). It is intended that this will enable third parties to create additional 'business intelligence' systems based on TrialVerse (E7). Moreover, the API can be used to build, test, and use automated systems for the extraction of data (E4). While the two user-facing systems have clearly separated concerns, they will be tightly integrated (E5), for example using:

- Single sign-on authentication
- Shared user and organization profiles
- Consistent visual identity

We also want to support corporate deployments of the ADDIS system (V2). This is possible because TrialVerse exposes a well-defined API. We can then do the following:

1. Deploy an internal system exposing company data through the TrialVerse interface (V3)
2. Deploy an internal version of ADDIS that has access to both TrialVerse and the internal database (V2)
3. (Optional) internal deployment of the analysis web services (V2)

To better support this scenario there should be clearly separated read-only (to be used by ADDIS) and read-write (to be used by the TrialVerse GUI) interfaces for TrialVerse.

4 Implementation strategy

The following is a possible transition strategy:

1. Implement the analysis web services based on R
2. Implement the TrialVerse read-only API based on the current ADDIS XML and a rudementary version of ConceptMapper
3. Build the ADDIS system (database and user-interfaces for meta-analysis and benefit-risk analysis, on par with the current ADDIS system)
4. Transition to a RDBMS-based system for TrialVerse (D5) – now ADDIS could already be a viable business intelligence system based on data we provide in TrialVerse
5. Start building a read/write version of TrialVerse and a more complete ConceptMapper
6. Integrate with TrialMine (E2)

By focussing on analysis functionality first, we have the quickest path to a useful system and enable the development of analysis functionality needed for several research proposals. In the interim, the desktop version of ADDIS can be used for data entry.