CHEMCONNECT: Intelligent repository backed by a knowledge base combustion related concepts

Edward S. Blurock[1]

1. Blurock Consulting AB, Lund, Sweden

INTRODUCTION

CHEMCONNECT is a smart cloud-based repository of experimental, theoretical and computational data. CHEMCONNECT goes beyond traditional data repositories in that the data is parsed and analysed with respect to an extensive chemical and combustion knowledge base. The parsed data is then linked allowing for efficient searching and comparison of combustion data. The goal is to have all data associated with experiments, from a device description, to the intermediate data (both computed and measured) and to their associated interpretations and the procedures and methodologies to the final published results and references to be available. Having published data linked to its dependent measurements and constants, devices, subsystems, sensors and even people and laboratories provides an effective accountability and more confidence in the data. Data entry and availability can range from private user, to user defined consortia to general public.

KNOWLEDGE BASE DATA ENTRY PROTOCOLS

Through the knowledge base, CHEMCONNECT adds meaning to the data and puts data in a more general context. This sets the stage for more efficient search and comparisons with other 'similar' data sources and for the presentation of data, for example, translating to another format and using different units.

The data sources are parsed and interpreted with the aid of a data *Protocol*, defined by the producer of the data source based on and supplemented with the CHEMCONNECT knowledge base. The protocol, as the name implies, specifies which parameters, measurements and results are expected to be reported. Templates for standard protocols, derived from domain researchers and standards within research communities, are defined within the CHEMCONNECT knowledge base. A protocol essentially defines which data is to be reported, how to interpret this data on a more general level and where to file this data in the data sources. So along with the storing the source data as in a typical repository, CHEMCONNECT 'applies' the protocol to the data source to parse and interpret the source data, making it more efficient to search (making a connection to more general concepts), display (more explicit knowledge of what the data looks like) and translate (for example, output in other formats).

The basis of the protocol is the standard reporting procedure defined by the laboratory, principle investigator or device community. These formats are usually based on convenience and historical reasons. Though recently, specific communities, usually connected with instrumentation and measurement techniques, are deciding within themselves standardizations of what constitutes 'good' or adequate reporting. The first task of the data protocol is to isolate blocks of data within the data sources. For example, if a data source is a matrix of data (a spreadsheet), several blocks of data could be reported in the same source. The protocol specifies how these blocks should be isolated.

The second task of the protocol is to identify the parameters that are reported in each of these blocks of data. An important aspect of this task is the identification and specification of units.

corresponding author: Edward S. Blurock, edward.blurock@gmail.com

In the template of the protocol from the knowledge base, the expected parameter has the class of unit, for example, time, temperature, pressure, distance, species amount, etc. When the protocol is defined for a data source, the actual units of the values are specified, for example, milliseconds, Kelvin, bar, centimeters, mole fractions, etc.
In practice, the protocol is defined once and then used as an efficient automated way to interpret data sets as they are produced by a given laboratory, principle investigator or community. Several types of protocols are available within CHEMCONNECT giving more information about the flow of data from initial measurements to final results. There are calibration protocols, specifying data associated with the calibration and setting up of the measurement device, for example, determining the species assignments in a mass spectrometer. There are measurement protocols for the production of the initial measurements. These protocols have an intimate connection with the device and sensor specifications. There are interpretation protocols that transform and manipulate the initial 'raw' measurements to intermediate and final results. For example, there are several stages of data manipulations in going from the temperature measurements in a heat flux burner plate to the final flame velocity determination. There are also reporting protocols giving templates for the 'standard' reporting of final results. The templates for these protocols come from domain knowledge extracted from publications, typical data source examples and the researcher's domain knowledge.

KNOWLEDGE BASE FOR DEVICE SPECIFICATION

CHEMCONNECT recognizes that an integral part of data measurement and reporting is the origins of that information, namely the device. This is particularly important when comparing results from 'similar' devices. Device and methodologies of using the device in text form within a publication or a publication that is referenced is not conducive to automated analysis and hinders, unless for the direct comparison within a publication, human analysis. To promote interconnectivity between measurements, devices and people making measurement using those devices, CHEMCONNECT allows for device specification. CHEMCONNECT views a device from a 'systems' point of view, i.e. that a device is composed with a set of subsystems (which could also consist of subsystems) and components (for example, sensors). In addition to the hierarchical view of the device, within each subsystem definition is a set of descriptive parameters and links to people, organizations and protocols.

OUTLINE OF DATA ENTRY

CHEMCONNECT distinguishes itself from other repositories in that it has a domain knowledge base that interprets the repository data. In a classical repository, a data file is uploaded, possibly with keywords and references. The file itself is left to the researcher to interpret.
In contrast, CHEMCONNECT uses a knowledge base to parse and interpret the data file so the data can be available for searching, visualization, interpretation and translation. The key to this interpretation is the protocol which specifies which data is to be reported, how the data is read in and how it is to be interpreted. As the name protocol implies, it specifies the framework for standard data reporting.
Data entry using the protocol in CHEMCONNECT has basically two stages: Setting up the specific specifications within the protocol and interpreting a data source as new measurements

1. **Protocol Setup:**
   a. Data Block Recognition: Specification of a 'block' of information within a data file (Data: **Block Definition** and **Isolated Matrix Object)**

b. Parameter Specification: How the parameters within the data should be interpreted, for example, which keywords are used for the parameters and the units of the parameters in the particular dataset. Correspondences can be made with other data using different keywords and different units. (Data: **Observation Correspondences)**
c. Protocol Specification: Typical datasets contain several sets of data that are defined by the experiments and devices. What information is (should be, as defined a 'good' reporting by the community) reported for a give experiment on a given device or what data sets are normally reported for a given laboratory is specified in a protocol. (Data: **Observation Specification)**

2. **Data Entry:**
   a. File Staging: How to interpret the file (formats, etc.) (Data: **Full Matrix Object)**
   b. Data entry: Using the protocol specification, the data files are interpreted, and the data is entered in the database. (Data: **Single Database Observation).**

The setting up of a protocol represents the preliminary, one time, work to set up the interpretation of (standard) data files originating from a given researcher, laboratory or community. The assumption is that file formatting does not change significantly with these groups. As new experiments are performed, data entry is repeated.

PROTOCOL SETUP

In preparation for reading in a typical set of data, a **Protocol** is selected (catalog data types are bold face). A protocol mainly specifies which data (observations) should be in the data source(s). For example, the standard reporting for ignition delay times from a rapid compression machine, *RCMPublishedResults* (knowledge base entities are italics), lists a set of expected observations, the fuel composition (*MoleFractionComposition*) and a set of parameters about the pressure and temperature (experimental and compression) of the run and the ignition delay time with uncertainty (*RCMPublishedResults*). Within the protocol, these are links to **Observation Correspondences**, giving the correspondence between the parameters in the source data, for example the link between the column of a matrix and a particular parameter. Each **Observation Correspondence** links the set of parameters in the source to each parameter in the **Observation Specification**. The Observation Specification has the type and unit information about each parameter. For example, in the expected observations from a rapid compression machine (*RCMPublishedResults*) the ignition delay time parameter type (*IgnitionDelayTime*) is specified. This ignition delay time specification from the knowledge base specifies that the units are of the class time (*TimeParameter*). The specific instance of the observation specification corresponding to the data source further specifies that the unit of time should be *MilliSeconds*. The **Observation Correspondence** also links to a specific **Block Definition** which specifies how the block of data is to be isolated from the source data from a **Full Matrix Object**. The result of the isolation is a **Isolated Matrix Object**. The Protocol, the Observation Correspondence and the Observation Specification could/should have additional links to the actual **Device**, for example on which device or which component on the device the data came from, to **Organization**, for example where the data is produced, and to **Person**, for example who the principle investigator is or who is performing the experiments.

DATA ENTRY

The **Single Database Observation** is the result of the source file (for example read into a **Full Matrix Object**) and interpreted by a **Protocol**. The **Protocol** (along with its associated entities and connections to the database) is set up once and is used to interpret all the data coming from

the source. The philosophy here is that laboratories usually have a standard or consistent format that they use for all their data storage. The **Protocol** describes this format. Each time an experimental set is performed, a source file is produced. This source file is read into the repository to a **Full Matrix Object** and then this, in turn is automatically interpreted by the appropriate **Protocol**. The isolation of the data and correspondence to repository data is automatically done by the definitions and specifications set up in the **Protocol** definition. The result is the **Single Database Observation**. The information in the **Single Database Observation** allows intelligent search through the interpretations provided by the knowledge base.

DATA AND KNOWLEDGE INTERACTION: THE HEART OF CHEMCONNECT

At a conceptual level, CHEMCONNECT consists of the interaction of three levels of data and knowledge:

- **Knowledge Base**: A representation (through an ontology) of the knowledge related to experimental devices and data. This knowledge base gives information on how data should be read and how to present the information. The knowledge base provides generic templates for specifications and entities.
- **Specifications and Entities**: Specifications build upon the knowledge base and provide details about how the source data should be read in and interpreted. Entities can be, for example, devices or subsystems and components of a device, researcher information or organization information.
- **Source data**: This is the actual repository data supplied by the researcher.

The **Knowledge Base** provides the generic knowledge and structural patterns on which specific instances catalog **Specifications and Entities** can be set up and entered in the database. For example, the Knowledge Base steers and sets up the user interface so the generic values can be replaced by specific values. The information in **Entities**, such as devices, organizations or people, are filled in, stored in the repository and represent information about the specific devices, organizations and people. The data of **Specification**s steer how **Source Data** is interpreted by given specific information about, for example, units or correspondences to **Source Data**.

For example, within the knowledge base, there is a generic description of a heat flux burner (*HeatFluxBurner*, a specific example of **Entities**) made up of a hierarchy subsystems and components, for example, the fuel preparation, burner, cooling system, thermocouples, burner plate, etc. (The knowledge looks at devices from a 'subsystems' point of view). In addition, there are set of parameter descriptions (suggested by researchers in the field) which differentiate different heat flux burners from different laboratories (for example, burner plate diameter, mass flow controller types, thermocouple types, experimental ranges, etc.). There is also a connection with type of observations come out of the device.

The knowledge base, for example, also sets up generic observations and parameters (**Specifications**). For example, an observation from the Heat Flux Burner plate (*BurnerPlateObservations*) is the matrix of thermocouple temperature measurements (*HeatFluxBurnerThermocoupleTemperature*) and distance from the center of the plate (*ThermocouplePositionInBurner*). To customize this to a given burner, the units of the generic temperature class (*TemperatureParameter*), for example, is set to specific units of celcius (*DegreeCelcius*) and units of the generic distance class (*LengthUnit*) is set to specific units of centimeters (*Centimeter*). This **Specification**, along with the correspondence between the **Source Data** parameters (viewed a matrix with the distance and temperature being columns) and the **Specification** parameters, is used to interpret the **Source Data**.