5.5.2.3. Supporting other data formats and data delivery methods

One of the greatest benefits of a dictionary-based informatics infrastructure is the flexibility that it provides in supporting alternative data formats and delivery methods. Because the data and all of their defining attributes are electronically encoded, translation between data and dictionary formats can be achieved using lightweight software filters without loss of any information.

XML provides a particularly good example of the ease with which data can be converted to and from the mmCIF format. XML translations of mmCIF data files are currently provided on the Worldwide PDB ftp site (ftp://ftp.wwpdb.org/pub/pdb/data/structures/divided/XML/). These XML files use mmCIF dictionary data-item names as XML tags. These files were created by a translation tool (http://sw-tools.pdb.org/apps/MMCIF-XML-UTIL/) that translates mmCIF data files to XML in compliance with an XML schema. The XML schema is similarly software-translated from the PDB exchange data dictionary.

Other delivery methods such as Corba (http://www.omg.org/cgi-bin/doc?lifesci/00-02-02) do not require a data format, as data are exchanged using an application program interface (API). A Corba API for macromolecular structure (Greer *et al.*, 2002) based on the content of the mmCIF data dictionary has been approved by the Object Management Group (OMG). Software tools supporting this Corba API (*OpenMMS*, http://openmms.sdsc.edu, and *FILM*, http://sw-tools.pdb.org/apps/FILM) take full advantage of the data dictionary in building the interface definitions and supporting server on which the API is based (see also Section 5.3.8.2).

5.5.3. Integrated data-processing system: overview

The RCSB PDB data-processing system has been designed to take full advantage of the features of the mmCIF metadata framework. The AutoDep Input Tool (*ADIT*) is an integrated data-processing system developed to support deposition, data processing and annotation of three-dimensional macromolecular structure data.

This system, which is outlined in Fig. 5.5.3.1, accepts experimental and structural data from a user for deposition. Data are input in the form of data files or through a web-based form interface. The input data can be validated in a very basic sense for syntax compliance and internal consistency. Other computational validation can also be applied, including checking the input structure data against a variety of community standard geometrical criteria and comparing the input experimental data with the derived structure model. The suite of validation software used within *ADIT* is distributed separately (http://sw-tools.pdb.org/apps/VAL/). All of this validation information is returned to the user as a collection of HTML reports.

In addition to providing data-validation reports, *ADIT* also encodes data in archival data files and loads data into a relational database. The loading of data into the relational database is aided by an expert annotator. The *ADIT* system customizes its behaviour according to the user's requirements. One important distinction is between the behaviour of the interface provided for depositing data and that of the interface used for annotating the data. The depositor is focused only on data collection and provides the simplest possible presentation of the information to be input. The annotator sees the detail of all possible data items as well as the full functionality of the supporting data-processing software and database system.

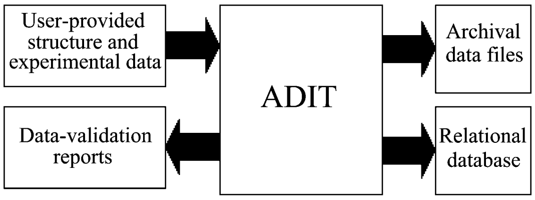

Fig. 5.5.3.1. Functional diagram of the *ADIT* system.

Although the *ADIT* system was originally developed to support the centralized data deposition and annotation of macromolecular structure data, it is not limited to these particular applications. Because the architecture of the *ADIT* system derives the full scope of information to be processed from a data dictionary, the system can transparently provide data input and processing functionality for any content domain. This feature has been exploited in building a data-input tool for the BioSync project (Kuller *et al.*, 2002). The *ADIT* system can also be configured in workstation mode to provide single-user data collection and processing functionality. This version of the *ADIT* system as well as the supporting mmCIF parsing and data-management tools are currently distributed by the RCSB PDB under an open-source licence (http://sw-tools.pdb.org/apps/ADIT).

5.5.3.1. *ADIT*: functional description

The basic functions of the *ADIT* deposition system are shown in Fig. 5.5.3.2. Users interact with the *ADIT* system through a web server. The CGI components of the *ADIT* system (that is, functional software components interacting with web input data through the Common Gateway Interface protocol) dynamically build the HTML that provides the system user interface. These CGI components are currently implemented as compiled binaries from C++ source code.

User data can be provided in the form of data files or as keyboard input. Input files can be accepted in a variety of formats. *ADIT* uses a collection of format filters to convert input data to the data specification defined in a persistent data dictionary. Data in the form of data files are typically loaded first. Any input data that are not included in uploaded files can be keyed in by the user. *ADIT* builds a set of HTML forms for each category of data to be input. At any point during an input session, a user may choose to view or deposit the input data. Users who are depositing data may also use the data-validation services through the *ADIT* interface.

Comprehensive data ontologies like the PDB exchange dictionary contain vast numbers of data definitions. A data-input application may only need to access a small fraction of these definitions at any point. To address the problem of selecting only the relevant set of input data items from a data dictionary *ADIT* uses a view database. In addition to defining the scope of the data items to be edited by the *ADIT* application, an *ADIT* data view also stores

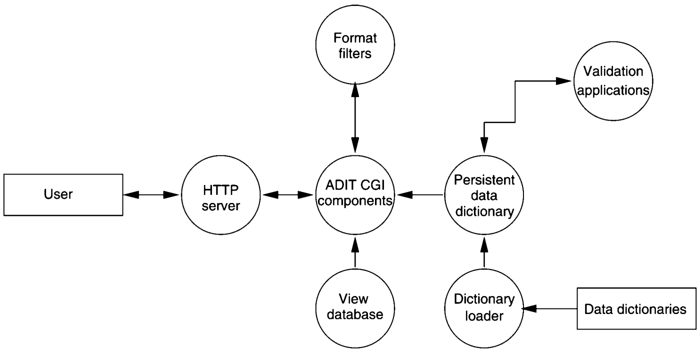

Fig. 5.5.3.2. Schematic diagram of *ADIT* editing, format translation and validation functions.