System Design and Implementation of the Integrated Sounding System

C. L. Martin and A.-L. Barrett
National Center for Atmospheric Research
Boulder, CO 80307-3000

The integrated sounding system (ISS) combines a suite of in situ and remote sensing systems with local computing capabilities. The resulting integrated instrument provides detailed real-time vertical profiles of the basic atmospheric parameters at a variety of time and space resolutions. One hallmark of the system is its configurability, which allows individual sensing systems to be added or removed as needed. A flexible data system copes easily with evolving and changing observation formats and requirements. A typical ISS configuration might include a wind profiler, a radio acoustic sounding system, a navaid based radiosonde, and a surface meteorological station.

Individual observing platforms in an ISS operate in a stand-alone mode and communicate via a common network to deliver high-level data products to the central workstation (Figure 1). The observing platforms often support archival of low-level data. Allowing the subsystems to operate independently prevents a fault in one system from affecting the others, and the low-level archives provide data redundancy.

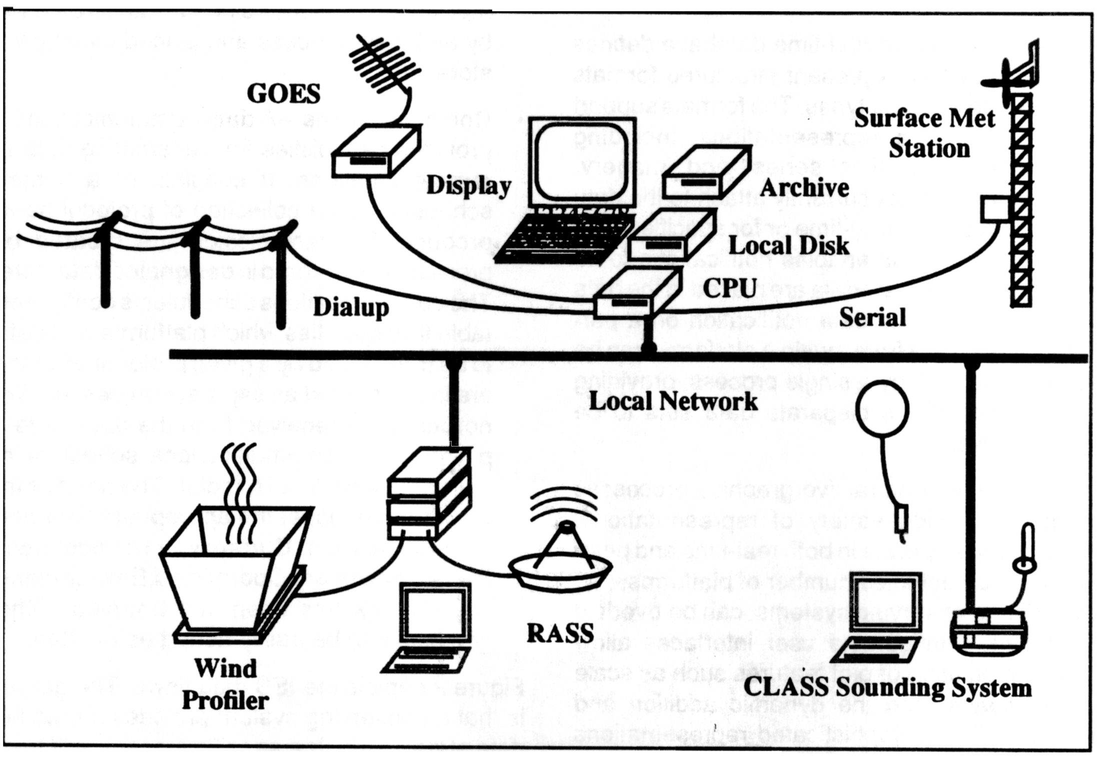

Figure 1. Typical ISS configuration.

The ISS workstation provides centralized resources to support a wide suite of activities, including system monitoring, data quality control, data archive, graphical display, analysis, and data product communication. Most procedures can operate with either the real-time data or the archived products. On-line storage of many months of observations is feasible in a typical ISS configuration.

The data system is implemented as a collection of cooperating processes. The NCAR Zeb data integration package provides the basic infrastructure supporting these activities. Basic facilities (both low and high level) include the following:

- Interprocess Communication—Message passing provides a generic facility for sharing structured information between processes. Capabilities are provided to multiplex message handling, other i/o, and compute activities. The message system is used to notify real-time processes when new data are available, coordinate between graphics activities, and synchronize processes within a subsystem such as within the data communications package.
- Data Store—A centralized real-time database defines "platform" objects which represent structured formats appropriate to different data types. The formats support a wide variety of data representations, including multidimensional grids, time series, and imagery. Multiple processes can concurrently attach to the data store and request data in real-time or for specified time periods. Real-time access enables notifications to be sent to processes when new data are placed in the data store. A process requests data notification on a per-platform basis, and data from multiple platforms can be received (or requested) by a single process, providing a mechanism that allows disparate data sets to be merged and analyzed.
- Graphics Processes—Interactive graphics processes display data in a wide variety of representations. Graphics processes operate in both real-time and post-time modes. An unrestricted number of platforms, i.e., data from differing observing systems, can be overlaid on a given plot. Configurable user interfaces allow interactive manipulation of plot features such as scale ranges, time spans, and the dynamic addition and removal of data fields. Sophisticated representations such as contours, vector time-series, and profile time-series are provided. Multiple graphics processes can simultaneously display in multiple windows, and graphics configurations can be opened and closed, so that a researcher can switch quickly between any number of configurations designed for that particular ISS deployment.
- Ingest Scheduler and Data Ingestors—A configurable process scheduler is provided for controlling data ingest activities and for scheduling other analysis and administrative tasks. Typically, the observing systems will place a data file containing an observation in a networked file directory, where it is noticed by the scheduler. The scheduler then will dispatch a data ingestor to format and transfer the observation to the data store. This scheme allows observing systems to be added to the ISS simply by constructing an appropriate ingestor (or using an existing one) and configuring the ingest scheduler to activate it. The ingest scheduler can also run processes on a periodic or continuous (i.e., "run once") basis. Ingestors are not limited to transferring observations from data files into the data store. Ingestors can access other data inputs; for instance, the surface met station produces a serial data stream which is read by an ingest process and placed directly into the data store.
- Communications—A data communications subsystem provides capabilities for transmitting data products to remote locations. It consists of a communications scheduler and a collection of protocol handlers. Data products for transmission are created by analysis processes and stored in designated data store platforms. The communications scheduler is configured through a table that specifies which platforms and data times are to be transmitted by a given protocol. Protocol handlers are implemented as separate processes. When a data notification is received from the data store for a given platform, the communications scheduler notifies the appropriate protocol handler. The handler then fetches the data product at the appropriate time and performs the transmission. Currently, a protocol handler to use the Geostationary Operational Environmental Satellite (GOES) link has been implemented. Other protocol handlers can be easily added as needed.

Figure 2 depicts the ISS data flows. The general scheme is that an observing system provides a data file (or serial data stream) which a specific ingestor will transfer to the data store. This ingestor is activated by the ingest scheduler.

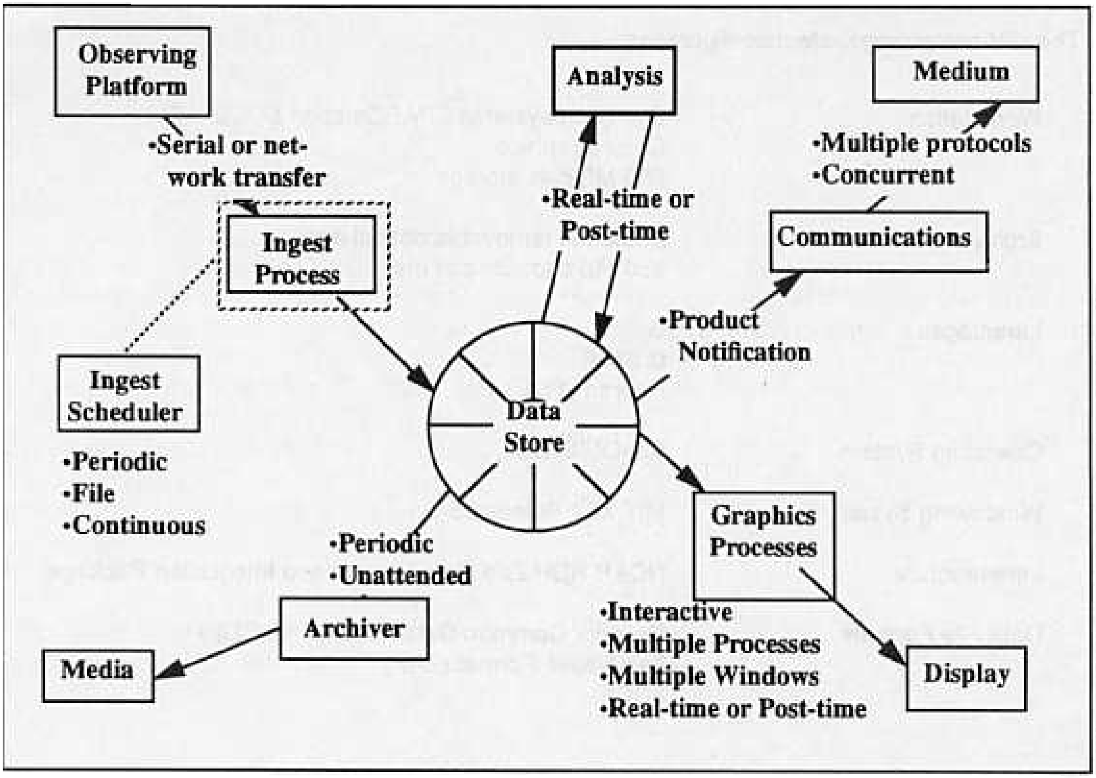

Figure 2. System data flows.

Real-time graphics processes receive notifications and then display the new data. Real-time analyses, e.g., a mesoscale model performing data assimilation, will also receive notification and will automatically obtain data as it becomes available. These models can generate products that are placed back in the data store and can be accessed by the graphics processes just like any other data. The data archiver is activated periodically to save the data store to permanent media. Analysis tasks are also activated to transform new data into products for transmission, and the communications scheduler notifies the protocol handler in order to initiate transfer of the product via the proper medium.

Table 1 outlines the current hardware and software foundation for the ISS computing system. This configuration allows for all source code and development tools to remain available on the system during an ISS deployment. The optical disk is used both for the permanent data archive and the system software distribution.

An ISS base station has been established at NCAR in Boulder to process data products transmitted from a network of ISS sites. Apart from an additional software subsystem which was implemented to reconstruct data products sent via the ISS GOES communications protocol, the base station uses the same software system that is employed in the field ISS. It concurrently processes data sets from the multiple ISS installations and provides for centralized real-time data collection, display and analysis of the observations coming from a multi-site deployment of the ISS. The sharing of software between the Boulder base station and the field systems has greatly reduced the software development and maintenance efforts required to implement and operate the central base station.

Table 1. The ISS computing system configuration.

Workstation	Sun Microsystems SPARCstation IPX, 24MB Color Graphics 850 MB disk storage
Archive	Erasable, removable optical disk 650 MB capacity per disk
Languages	C C Shell Fortran (1%)
Operating System	SunOS UNIX
Windowing System	MIT X11 Release 5
Infrastructure	NCAR RDP Zeb Data Display and Integration Package
Data File Formats	Network Common Data Format (NetCDF) Zeb Native Format (ZNF) ASCII