

MC3-to-HSII Data Interface (Update Pathway) MIS Executive Overview

William Hanna, Project Leader

Prepared by: Jeff Szuhay
KHP Services, Inc.

February 6, 1995

This document is proprietary and confidential information of Pennsylvania Blue Shield and KHP Services Inc.

It is not meant to be distributed or released to non-PBS nor non-KHPS employees.

Introduction

This is intended to provide a *brief* summary of the MC3-to-HSII Data Interface, often referred to as “on-line transactions.” Please note that other data pathways between MC3 and HSII data repositories are not discussed.

This document assumes general familiarity with KHP Service’s MC3 application, HSII application, as well as certain data communications concepts. Other documents exist to provide more detail; please see the “For Further Reading” section at the end of this overview.

MC3-to-HSII Data Interface

The MC3-to-HSII data interface focuses primarily on synchronizing “update” transactions initiated on the MC3 “front-end” application with the HSII “back-end” system. The MC3 front-end is built with PowerBuilder. The HSII back-end runs in a MUMPS environment on Unix.

Currently, the HSII system has been designed strictly from an interactive user’s perspective. Therefore, our general approach is a kind of screen scraping—acting upon/reacting to data streams from HSII screens—not typical screen scraping. When HSII provides methods for automatic updates to their data (either via EDI interfaces or externally callable routines), these methods will replace our screen scraping approach.

Given this, some key assumptions were made in the design and implementation of this data pathway which must be recognized before customer-driven changes can be proposed:

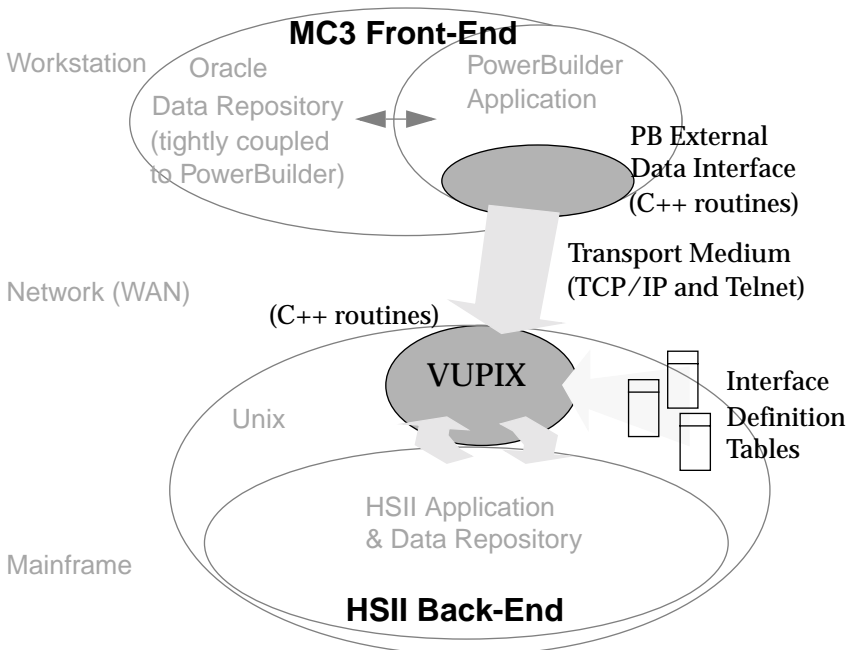
- the MC3 application is tightly coupled to a particular HSII version (and screen set). The MC3 programmers must be keenly aware of the HSII screen set(s) to be updated for a given transaction. From the MC3 user’s perspective, however, there is little or no correlation of user interaction between the two systems.
- Both systems, MC3 and HSII are subject to both general changes as well as customer-specific changes. Such changes are driven by the dynamics of the managed care industry. Further, site-specific configuration parameter changes (not considered program modifications but a regular part of the HSII application) can cause unexpected changes in assumed screen behaviors.

This data interface consists of 4 major parts:

1. an interface from the Powerbuilder application to the transport medium (Powerbuilder-to-network). This interface is a set of function calls known to Powerbuilder which format and transmit its data to a set of network-aware routines external to Powerbuilder.
2. a transport medium from the MC3 workstation over a wide-area network to the HSII mainframe. This consists of TCP/IP utilizing Telnet sessions. It is compatible with Novell, Vines, and LAN Manager; however, none of these is required for its operation.
3. an intermediate process between the transport medium and the HSII application which, upon receiving transmitted data, presents input to, and deals with output from the HSII system. This is known as the

VUPIX or Virtual User Process Interpreter Translator (because it acts like a user typing in data; it runs as a process independent from MC3 and HSII systems; and converts and “knows” about incoming data and expected outputs from both MC3 and HSII systems).

4. a set of supporting “interface definition tables” which store knowledge about both incoming data from MC3 and system interaction (or screen knowledge) with the HSII system. These tables are intended to make the system data-driven and highly responsive to changes in either the MC3 application or the HSII application without necessarily demanding the skills of a programmer.



Notes on Each of the 4 Parts

The PowerBuilder-to-network interface consists of a set of internal PowerBuilder function calls (function definitions) and a set of external C++ routines on the MC3 workstation (these routines exist on the same machine as the Powerbuilder application).

The transport medium is intended to be “network neutral” and currently employs TCP/IP (which requires no additional file or communication

server or NOS). It also relies upon a Telnet session with the Unix host, providing Unix-level security and ease of administration.

The VUPIX is a set of C++ routines which run on the Unix mainframe and interact directly with a MUMPS shell/HSII application. Each PC workstation has its own VUPIX session. Execution of a given user's VUPIX session is controlled via configurable login scripts.

The Interface Definitions Tables are maintained by a separate PowerBuilder application in a separate data repository (currently Watcom SQL) and are uploaded to the Unix system at controlled maintenance points. The PowerBuilder maintenance program enforces table data integrity and consistency while providing a friendly user-interface to the maintainers.

It is intended that, once developed, the VUPIX, the transport medium, and the data structures will remain fixed and relatively immutable. Most modifications will require table maintenance with possible changes to the PowerBuilder-to-network interface.

The design of the VUPIX, while currently implemented specifically for the MC3-to-HSII interface, is general to client/server requirements and modular in structural design so that it may be applied to alternate front-end applications (e.g. rapid claims entry, Key Link, etc.) as well as alternate back-end systems (e.g. Tingley, AIH, etc.).

For Further Reading

Please contact Bill Hanna (760-9059) for the most current edition of each of the documents listed below.

HSII / MC3 Data Pathways Overview by Jerry Sandridge

HSII / MC3 Overall Design Specification by Jeff Szuhay

VUPIX General Design Specification by Jeff Szuhay

Interface Function Definitions by Bill Hanna (the Powerbuilder function call definitions).

VUPIX Tables General Design Specification by Jeff Szuhay

VUPIX Implementation Overview and Code Specifications by Jeff Szuhay