Whole Person Integrated Care (WPIC): A Healthcare Transformation Strategy Supported by a Novel Spreadsheet-Based Software Framework

Stephen E. Beller, PhD  
National Health Data Systems, Inc.  
Croton on Hudson, NY USA  
sbeller@nhds.com

Sabatini J. Monatesti  
Senior Member, IEEE  
ES Enterprises, Inc.  
Berwick, PA USA  
smonatesti@verizon.net

Peter Bachman  
Ceqs, Inc.  
Swathmore, PA USA  
peter.bachman@pahisp.org

Abstract — This paper describes a humanitarian healthcare transformation strategy called Whole Person Integrated Care (WPIC). WPIC couples a Spreadsheet-based Software Framework (SSF) and a care delivery model that focus on increasing efficiency and effectiveness through evidence-based knowledge and action. This strategy: 1) integrates “sick care” (treatment of health problems) with “well care” (prevention and self-management) and 2) addresses both physical and psychological problems affecting a patient’s health. The SSF currently utilizes Microsoft Excel to create simple, low-cost, and flexible software solutions that capture, import, export, transform, integrate, organize, store, analyze, render, and exchange data and information. Using a node-to-node (point-to-point) network architecture, clinicians, researchers, patients, and others can connect securely and resiliently, anywhere and anytime, across national and organizational boundaries. This allows collaborators to build, share, evaluate, and evolve evidence-based knowledge and analytic models. The shared knowledge and models are used to make projections, reveal existing problems, identify risks, and offer solutions that support decisions for clinical and business process improvement. The paper also describes an SSF application recently piloted in a Federal health information technology program, and proposes an open source community.

Index Terms—analytical, biopsychosocial, collaboration, data exchange, decision support, framework, health information technology, healthcare, models, network architecture

I. INTRODUCTION

This paper focuses on the use of a novel spreadsheet-based software framework (SSF) to create applications (SSF apps) designed to: 1) manage and share evidence-based knowledge focused on process improvement decision support among collaborators connected in loosely-coupled social networks and 2) realize the humanitarian potential of the Whole Person Integrated Care (WPIC) strategy. A document showing and describing examples of current SSF apps is available at: https://www.nhds.com/app/download/766307452/SSF+Apps.pdf.

A) WPIC Strategy

The WPIC strategy utilizes health IT knowledge models to integrate “well care” with “sick care, and to support holistic care that addresses psychological and physiological problems, and the interactions between them.

Well care attempts to prevent illness and dysfunction through healthy lifestyles and through effective self-management of chronic health problems. Sick care provides medical and non-medical treatments to people who ill, distraught, or impaired.

As described later in this paper, WPIC’s goal is to:
• Lower overall healthcare costs by keeping people well longer and providing needed care economically and competently.

• Increase measurable care quality through widespread collaboration among clinicians, researchers, and others, across organizational and national boundaries, using a ubiquitous, scalable, low-cost technology that consumes little resources and has minimal technical requirements, i.e., a spreadsheet program, simple computer, basic e-mail, and occasional internet access.

B) Reimagining Spreadsheets

We present a reimagined view of spreadsheets as being much more than elaborate calculators. We show how they can be used to enable collaborative knowledge feedback loops and support decisions for global healthcare quality improvement.

C) Software Framework Defined

A software framework exposes universal or generic functionality that can be used to develop domain specific software through the execution of external, user-written code. Software frameworks may also include supporting code libraries, tools, application programming interfaces (APIs) and other utilities that enable development of a software project or system. In this way, a software framework provides a consistent, reusable, and standard way to build and deploy software applications, products and solutions.

D) Spreadsheet Program Defined

A spreadsheet program, such as Microsoft Excel, produces spreadsheet workbooks. Each workbook contains one or more worksheets. A worksheet is a grid of cells organized in rows and columns. Each cell is a “computational container” that stores and control (calculate, modify) numbers, text, formulas, functions, hyperlinks, and images. The cells can be populated by: 1) manual user input (type or paste data, links, and images into a cell); 2) data imports; and 3) user forms.

Spreadsheet workbooks may also: 1) contain macros, i.e., programming code modules that automate software processes such as Excel’s VBA (Visual Basic for Applications); 2) include add-ins, i.e., tools that adds new features, functions, and efficiencies to a workbook; and 3) utilize third-party software.

E) Spreadsheet-based Software Framework Defined

A SSF is a software framework that supports the creation of SSF apps. These applications provide a low-cost, efficient, and less complex way to capture, import, export, transform, integrate, organize, store, analyze, and render data. They also have qualities that make them ideal for the creation, exchange, and management of multifaceted software models that defines an application’s capabilities (features and functions).

The paper includes discussions of: 1) how the SSF emerged; 2) the capabilities, advantages, and risk mitigation of current SSF apps; and 3) the SSF’s potential for innovation that
improves human health and well-being through the growth, transfer, evolution, and use of evidenced-based knowledge.

The origins of the SSF’s are discussed next.

II. INITIAL APPLICATION DEVELOPMENT

Dr. Stephen Beller started his professional career in 1981 as a licensed clinical psychologist. That same year he was introduced to spreadsheets and began to think about novel ways of using them to help improve his psychotherapy services. This was the beginning of a 30-year journey that led to the creation of the novel SSF.

A) Development of a Spreadsheet-based Application for Mental/Behavioral Healthcare

Dr. Beller first developed and tested a spreadsheet-based psychological questionnaire to collect and analyze copious information about his patients. He used that information for treatment planning and to assess treatment outcomes. He then extended it to include the following types of data:

- medical data since physical health often affects one’s psychological health and vice versa [1];
- pharmaceutical data since drug side-effects can cause problems with mood, cognitions, and behaviors [2];
- information about treatment methods, procedures, and tests to enable researchers to establish clinical guidelines for patient diagnose and treatment; and
- patient satisfaction ratings.

Some of the questions were designed to be answered by the patient, while others were answered by the clinician. Over the next few years, after extensive literature review and field testing, he added about 3,000 new questions and organized them into about 300 biopsychosocial categories. Each question includes metadata, i.e., an ID number, response scale (i.e., quantitative metric units used to answer the questions), and question skip-logic (branching-logic) rules that present only questions relevant to a patient based on previous answers.

B) Transformation into a Health Information System (HIS)

The psychological questionnaire was transformed into a comprehensive SSF HIS designed to enable clinicians (doctors, nurses, therapists, diagnosticians, etc.) to deliver healthcare services as efficiently, effectively, and safely as possible by building and transferring evidence-based knowledge. The specific purpose of the SSF HIS is to:

- provide essential health information to all types of clinicians, not just psychotherapists;
- deliver patient and treatment information securely to researchers and subject matter experts who develop and share clinical guidelines (evidence-based best protocols, advice, recommendations) that support clinical decisions and actions;
- enable collaboration between the clinicians who treat the patient (i.e., the “care team”) to provide coordinated care and continuity of care;
- promote shared decision-making whereby clinicians educate their patients about health problems, risks, and care options with consideration of the patients’ individualized circumstances, and preferences [3].

C) Interacting Aspects of the SSF HIS

Fig. 1 is an overview the SSF HIS’s interacting aspects, including stakeholders who collaborate to develop and use SSF apps focused on building and implementing knowledge for continuous improvements in healthcare processes that would bring greater value to patients/consumers through education, research, and evidence-based decision support.

III. SSF HIS’s SEVEN FUNCTIONAL REQUIREMENTS

To achieve these objectives, SSF HIS apps need seven functional requirements; they must be able to:

1) Collect and store vast amounts of diverse, evolving data for each patient.
2) Query local and remote data stores, including databases, data marts, and data warehouses.
3) Be standards agnostic.
4) Perform data transformations and analyses to generate reports that build useful knowledge and insights.
5) Exchange data, reports, guidelines, and conversations —efficiently, at low cost, across the globe—in secure and resilient networks using any architectural style.
6) Provide intuitive user interfaces that accommodate clinical and business workflow processes.
7) Continually adapt to new knowledge and techniques.
We will now discuss how the seven functional requirements of the SSF HIS have been satisfied.

IV. SATISFYING THE FUNCTIONAL REQUIREMENTS

The SSF HIS satisfies the functional requirements through the novel use of Excel.

A) Requirements Satisfied with Excel Workbooks and Add-ins

The SSF leverages the native (built-in) capabilities, features and functions of Excel workbooks and add-ins to develop the SSF HIS (and other) applications. These capabilities include numeric calculations and string manipulation; process automation and rules logic; data sorting, filtering, and grouping; table and list management; database queries; data transport, imports and exports; networking; data cleansing, validation, transformation and integration; data storage, retrieval, encryption and decryption; spreadsheet template management; desktop and cloud computing; API access; user form creation and implementation; hyperlink use; presentation (dashboard development, data visualization, chart creation, page layouts, conditional formatting); integration with third-party software; and more. Excel add-ins increase its capabilities. More information is available in books and web sites [4], [5], [6].

B) Requirements Satisfied with Excel VBA Macro Modules

VBA macro modules control the way these new applications used the components, processes and rules of previous applications. These modules consist of programming code organized in interacting subroutines (sets of instructions software programs use to perform specific operations).

C) Requirements Satisfied with Spreadsheet Models

Each application is based on a multifaceted spreadsheet model. Each of these models use algorithms—comprised of mathematical, logical, formatting, and procedural rules—to: 1) obtain, analyze, and present data that simulate real-life situations designed to provide useful feedback and insights; 2) accommodate effective workflow processes; and 3) provide a high level of software usability. That is, a model defines an application’s capabilities and actions.

The facet of a spreadsheet model that determines an application’s UI, for example, defines its appearance and what a user must do to make the software operate. Other parts of the model define how the application accommodates user workflow processes, e.g., what data to obtain, where and how to obtain them, what to do with them, where to store and send them, and how to present them.

A spreadsheet model’s analytical facet, on the other hand, defines how to analyze the data and present the results. In healthcare, these models may be designed to support clinical and business decisions by “synthesize[ing] evidence on... from many different sources...[and by providing] a logical mathematical framework that permits the integration of facts and values and that links these data to outcomes that are of interest to...decision makers...[T]he end result of a model is often an estimate of...value for money...[The models should also be],...subjected to thorough internal testing and ‘debugging’...[and] peer review...If a model’s outputs differ appreciably from published or publicly available results based on other models, the modeler should...explain the discrepancies...[and] cooperate with other modelers in comparing results and articulating the reasons for discrepancies...Models should be based on the best evidence available at the time they are built...[and] adapt to new evidence and scientific understanding...They should be repeatedly updated, and sometimes abandoned...as new evidence becomes available.” [7].

Ideally, the data used by healthcare models should be integrated from disparate sources—across a patient’s entire lifetime—to promote integrated, coordinated care. The applications that use such analytic models enable patients and healthcare professionals to make evidence-based judgments [8].

D) Requirements Satisfied with the Operating System

Our SSF HIS and other SSF apps also leverage capabilities of the operating system’s APIs, components, and services. These capabilities enable user input and messaging, data access and storage, system diagnostics, applications and device handling, operation of devices, system services, security and identity, application installation and servicing, system admin and management, internet use, and networking [9].

1) Exchange Data in Node-to-Node Networks

SSF apps use operating system’s services and messaging to enable the exchange (transport) of data. This can be done in encrypted e-mail attachments, for example, that are sent asynchronously in secure node-to-node (point-to-point) networks. A node, in this case, is a spreadsheet application with both sender and receiver transport capabilities.

We typically use encrypted e-mail attachments for transport data via the node-to-node networks because it is Internet standards based, convenient, inexpensive, and ubiquitous for mobile and desktop applications. We implement SMTP with S/MIME [10] since it has known threat models and well-established methods of security. This security is maintained through end-to-end encryption to prevent data from being modified en route and at rest.

Benefits of these networks include: 1) support for loosely-coupled social networks and 2) network resiliency and extended network reach through satellite internet.

a) Loosely-Coupled Social Network Support: Because our applications use e-mail to connect people in node-to-node networks, they can communicate with one another any place and any time without constraints imposed by a central authority. Such “loosely coupled” social networks (LCNs) enable diverse groups of people to communicate (exchange information and transfer/share knowledge) and collaborate (work and learn together) through pre-established or ad hoc (impromptu) network connections. The LCN e-mail communication architecture is an easy, low cost, and secure way to connect individuals anywhere in the world in networks that cross organizational, professional, political, and geographical boundaries. As such, they connect a greater variety of people than tightly coupled (narrow/closed) networks. These LCNs, in turn, increase innovation and creativity, promote collaborative model-building, and enable cross-national learning:

- LCNs include connections between people who are “loosely connected,” i.e., they do not belong to the same social groups, organizations, etc. Innovation and creative solutions are fostered through the “strength of weak ties.” Unlike strong ties—characterized by
homogeneity of experience and knowledge, access to the same redundant information and resources—weak ties supply diverse sets of data, experience, education, expertise, knowledge, ideas, perspectives, and other relevant, non-redundant information and insights [11]. Thus, LCNs help spark innovative ideas and creative strategies. This approach was developed originally—in contrast to guilds that used protected knowledge—to promote “useful knowledge” between loosely connected peers as a form of hybrid vigor [12].

- LCNs enable diverse groups of collaborators to build, share, evaluate, and continually improve software models, the importance of which was discussed previously.

- LCNs support cross-national collaboration, such as studies of fundamental healthcare issues, as well as managing worldwide epidemics. This is important because “...international comparisons create and sustain the impetus for change...having examples of countries that excel shows us what is possible...Fortunately, international research has the potential to generate insights into how to manage shared problems...[that] include: finding optimal methods for treating chronic conditions, coordinating care...and ensuring that care is affordable and respects patients’ values and preferences...Importing and adapting...innovative tools and practices from our international peers can improve our performance [and] exporting the lessons we learn...may help improve outcomes and experiences abroad” [13].

b) Network Resiliency through Radio Internet and Extended Reach through Satellite Internet: SSF apps use a “store and forward” transport neutral method in which a sender stores data in SSF-based data files, as well as in on- and off-premises data stores if desired. A data file (DF) provides a very efficient way to store large amounts of data (see Appendix). The DF is sent via encrypted e-mail to one (or a series of) intermediate stations where it is stored temporarily before it is sent (forwarded) to its destination. If the sender is connected to a satellite, then the e-mail can be sent via a satellite Internet [14] wherever and whenever desired, and the receiver can retrieve it at any time. The same is true with radio service connections. This means:

- Network connections are resilient since satellite and radio Internet communications are available when ground based Internet connections are not available. Furthermore, since a DF can pack a great deal of data in a small file, bandwidth constraints or expensive transmittal costs become less of an issue.

- Network reach is extended across the globe, even to less technology adapted civilizations several hundred miles from the nearest DSL service provider or just simply whiling their time in an island away from the rest of the world. Satellite internet is useful in remote places because it is easy to access any time and place. Africa is a good example [15]; Microsoft is using TV White Space to aid connectivity [16].

To enjoy the benefits of satellite internet, every user must have the appropriate equipment: a fast computer, modem, satellite dish, ISP, and an unobstructed area.

2) Transform, Translate, and Integrate Data: SSF apps have access to a universal translator (UT) that adjusts a model’s data, semantics (terminologies/vocabularies), formats, and visualizations as they are shared by different people in different organizations, departments, social groups, and regions. The UT, which is a component of the SSF’s DTW, provides a means to modify (transform, translate) information as it passes between people to ensure everyone receives the right information, in the right format (structure), and with the right semantics. This helps increase productivity by reducing misunderstandings.

After any necessary transformations and translations are completed, data are cleaned [17], integrated, and aggregated, so they are ready for analysis and presentation.

F) How the SSF HIS has been Implemented to Satisfy Each Functional Requirements

This section discusses how the novel software methods above are used to build SSF HIS and other SSF apps that satisfy the seven functional requirements.

1) Collect and Store Vast Amounts of Diverse, Evolving Data for All Patients: This requirement is satisfied by our SSF HIS’s capacity to encompass all patient health data.

The questions and their metadata are stored in CSV-formatted “inquiry files;” the answers and their ID numbers are stored in DFs as previously discussed. The inquiry files are extensible: Items can be added, modified, or removed easily and new file versions are indicated by their file names and dates.

Each inquiry file contains the questions relevant to different use cases such as adult versus child diagnostics, clinician versus patient forms, and pre-test versus post-test assessments. Excel macros: 1) populate and load user forms with the questions in the inquiry files; 2) load the user forms, which present the questions in accordance with the skip-logic rules; 3) capture and validate the user’s responses; and 4) store the user’s inputs in the cells of the patient’s DF.

Note the non-HIS SSF apps use different data sets with similar computerized processes and components.

2) Perform Queries: Some SSF HIS DFs contain data about a single patient. The data sets from the DFs of other patients are stored as structured tables in other Excel workbooks. The SSF HIS can store data in a relational database tables as well. The SSF HIS used Excel macros to: 1) query these workbooks and database tables; 2) write the data from a query to worksheet in the SSF HIS application, and 3) use those data for analysis and presentation.

With the advent of Service Oriented Architecture (SOA), and more recently microservices, as well as the ever-increasing availability of APIs in recent years, the use of third-party middleware tools to facilitate queries and data transformations has become easier and more common. SSF apps can leverage these tools.

Excel’s query capabilities were recently expanded to include two add-ins: PowerPivot and Power BI Desktop. These free tools overcome Excel’s worksheet row limitations through an “in-memory data modelling component that enables highly-compressed data storage and extremely fast aggregation and calculation” and Power BI can connect “to on-premises and
cloud sources using a combination of direct query and periodic data refreshes” [18]. PowerPivot enables SSF apps to use complex criteria to query large databases [19]. Furthermore, a recent addition to Excel makes downloading, updating, and transforming data from the Web even simpler with the addition of a new built-in automated process.

3) Be Standards Agnostic: Through participation in several Federal health IT technical workgroups over the past five years, we have demonstrated our SSF HIS applications ability to comply with multiple standards including:

- Data format standards, e.g., XML, JSON, CSV
- Vocabulary/code set/terminology standards, e.g., for diagnoses, procedures, lab results, and more [20]
- Transport standards, e.g., SMTP, SOAP, and RESTful
- Performance standards that evaluate process compliance and outcomes.

To achieve standards compliance, SSF apps: 1) transform and extract data and metadata elements from delimited data formats using string parsing functions; 2) translate non-conformant data elements to standardized vocabularies, code sets, and terminologies using worksheet maps and Macros; and 3) avoid loss of semantic nuance when local terminology standards are replaced by a single global standard using its universal translator (described above).

4) Perform data transformation and analyses to generate reports that build useful knowledge and insights: SSF apps leverage Excel’s native analytic and presentation capabilities to create reports that build knowledge and insights that help improve processes and outcomes.

a) Analytics: An SSF apps data analysis module uses Excel’s native analytic tools use worksheet and macro formulas and functions, as well as its Analysis ToolPak add-in [21], to perform complex statistical and engineering analyses. The results of these analyses are stored in worksheets. The module can also use Excel’s what-if analysis tool (called Solver) [22] that finds an optimal (maximum or minimum) value for a cell formula. These computational capabilities satisfy many use cases, but additional statistical packages can be used as necessary [23]. SSF apps can use artificial intelligence to help solve exceptionally complex problems [24], [25].

b) Visualizations: All versions of Excel can create numerous type of charts and tables with user-interactive capabilities. In newer versions, data can be added to maps and represented in people graphs [26]. In addition, the PowerPivot and Power BI Desktop add-ins, introduced earlier, provide extensive visualizations and sharing capabilities [27].

5) Exchange Data Files, Reports, Guidelines, and Conversations in Secure, Resilient Networks Across the Globe: Our node-to-node network design, discussed earlier, is ideal for the secure exchange of data, information, and models that support ongoing research-based collaboration and care coordination among clinical team.

6) Provide User Interfaces (UIs) and Automated Processes that Accommodate Clinical and Business Workflows: Macros and user forms enable the applications to implement UIs that conform to the desired workflows at a “granular level.” That means every software operation must correspond to the appropriate steps in the current clinical and business workflow processes. All our SSF apps demonstrate this capability. The workflow process requirements are determined by subject matter experts and test implementations that identify: 1) what should be done; 2) how it should be done; 3) when it should be done; 4) where it should be done; and 5) who should do it [28].

7) Continually adapt to new knowledge and techniques: Information technologies, software models, and knowledge management processes evolve over time. As such, applications must change accordingly or become obsolete. The modular structure and the inherent flexibility of SSF apps have enabled them to evolve for decades; there is no apparent reason for this end.

V. SSF HIS APP CAPABILITIES AND ADVANTAGES

As mentioned earlier, emergence of the SSF HIS is due to its inheritance of components and processes in our diverse applications that satisfy the seven functional requirements. Most of the two dozen applications we have developed are part of our SSF HIS toolbox. They focus on the healthcare industry’s need for tools that provide secure and efficient information exchange, support clinical decisions, capture and deliver of data for research studies and performance assessment, and enable self-management of chronic medical health conditions and psychological problems. The non-healthcare specific (non-HIS) applications, on the other hand, focus on project management and sales, financial services, emergency response, social services, education, and natural resource exploration.

All SSF HIS apps have important technical capabilities that give them the following unique set of benefits and advantages. See the Appendix for a technical explanation and diagrams of the SSF application architecture.

A) SSF HIS Technical Capabilities

1) SSF HIS Workbook Capabilities: Worksheets, macros, and add-ins give the SSF HIS apps the ability to: 1) connect people easily and inexpensively in secure loosely-coupled social networks; 2) obtain, store, and share vast amounts of diverse evolving data efficiently and securely; 3) consume, parse, and transform data to, and from, any formats; 4) present data and information in a wide range of reports that can include user interactive dashboards and advanced visualizations; 5) secure data in transit and at rest through strong encryption; 6) comply with terminology standards without loss of semantic nuance; and 7) build, share, evolve, and implement analytic, workflow, and software usability models.

2) Reusability of Software Components and Processes: Reusability is another technical capability that is critical to the emergence and sustainability of the SSF HIS. It refers to the use of existing software components and processes in different SSF HIS apps. As such, each SSF HIS app we develop reuses parts of previous SSF HIS apps to satisfy certain functional requirements. These reusable components and processes are, in turn, inherited by the SSF HIS. It is typically easier and more efficient to modify reusable software parts for a new application than to develop new ones from scratch [29].

3) Permanence and Inheritance. These architectural artifacts enable SSF models to retain data when changes are
made (permanence) for possible later use (e.g., via audit trails, restore, and time-event correlations). In addition, a model’s attributes and methods can be inherited from other models, e.g., when similar queries are performed multiple times for different patients.

B) SSF HIS Benefits and Advantages

Based on the SSF’s technical capabilities, the benefits and advantages our SSF HIS apps include:

- Low barrier to entry, e.g., when an SSF is used with Excel and MS Office it provides extensive technical capabilities in an inexpensive, ubiquitous, and familiar commercial off the shelf (COTS) product.
- Time, money, and resource savings through an automated process by which the apps: 1) create, encrypt, address, send, and track e-mail message that contain data for transport; and 2) receive, decrypt, store, consume, and render that data. This process: 1) minimizes data transmission and storage costs, while consuming minimal bandwidth; 2) requires no infrastructural build-out; 3) has low overall complexity; and 4) connects people across the globe with ease.
- Increased innovation and more effective decision-making by enabling people in loosely coupled networks (LCNs) to: 1) share diverse experiences, data sources, information, knowledge, expertise, perspectives, ideas and insights; and 2) exchange and evaluate computerized models to improve their ability to supply useful, relevant, actionable information for performance improvement and other purposes.
- Reduction of technological cost and complexity by managing enormous quantities and diversities of data through a simple spreadsheet template and file management method.

C) Risks and Mitigations

End-to-end encryption, hardened devices, user data control, digital signatures for authentication and authorization, revision controls, and security risk analysis [30] help avoid problems with the security, privacy, and integrity of data, as well as the versioning of models and data files. Spreadsheet auditors (e.g., ExcelAnalyzer), neural network tools [31], automatic spreadsheet generators [32], unit testing, adhering to spreadsheet best practices, and testing against exemplars with validated results are ways to deal with (potential) modelling errors.

By way of example, we will now present an SSF HIS application that was recently piloted successfully in a Federal project and received positive reviews.

VI. SSF HIS HEALTHCARE IMPLEMENTATION EXAMPLE

The Structure Data Capture (SDC) project was initiated by the Federal government’s Office of the National Coordinator (ONC) in 2013 [33]. SDC establishes standards necessary for electronic health record (EHR) and other HIT systems to capture, store, transmit, and use structured data in the real world for clinical research, patient safety and adverse event reporting, public health reporting, and determination of coverage.

Dr. Beller is a member of the SDC technical workgroup that evaluates those standards. He is one of several participants who developed and piloted a “form filler” (FF) application that builds and implements electronic forms. Only he used the SSF.

A) Pilot Implementation Method

The pilot implementation follows these steps as required by the SDC project:

1. A user click a button on the UI and indicates the form to build from a list, which sends a SOAP message to a “form manager” web site requesting an XML file called a “form design file” (FDF). The FDF contains form-building instructions using a complex XML schema, with over 100 hierarchical elements and attributes, that define the appearance and behavior of each item (object) in a form. Many different forms can be built with this same standardized schema.

2. Upon receipt of the FDF, the FF application parses the XML and transforms it into a worksheet in a format that is very easily to read, navigate, and use. Fig. 2 shows a small section of an FDF that actually has 2211 rows of XML and its corresponding worksheet that actually has 417 rows data (one for each form object) and 107 columns that define each object’s properties (appearance) and behaviors (rules-based actions).

3. The FF uses the spreadsheet’s contents to build and display the form in real time. If a form contains questions that requires answers from a remote source, the FF retrieves the data and writes them into the form.

4. The FF writes all data input into the form to the spreadsheet and, once the form is complete, it embeds the answers in the FDF’s XML. It then transmits the data to other data stores (“form receivers”) selected by the user and creates a printable report.

Our FF implementation was successfully piloted by demonstrating that the data input into its forms were received without error by the form receiver.

B) Example of an SDC Form Filler

Dr. Beller used the SSF HIS app FF to build multiple user forms with ActiveX controls (form objects) [34]. A portion of one such form appears in Fig. 3. The form’s name and ID (upper-left) indicates it collects data (e.g., from pathologists) about adrenal gland cancer. Fig. 3 identifies a few form objects types; the numbers identify the objects described below:
1. Section header that indicates the questions are for the clinical concept called “Accessory Findings.”
2. Question object (QO) for which only one answer-objects (AO) may be selected from its three-item single-select answer set.
3. AO that has not been selected. The FF colored it green after the AO beneath it was selected; green means the QO has been answered with an AO from its answer set, so it is in a “confirmed false” state. Note that before the user selected it, all three AOs were yellow to indicate the question is optional and need not be answered.
4. A selected AO that the FF formatted with bold text and a black dot to the left, which indicates the AO was selected and is now in a “true” state.
5. One of four sibling AOs (i.e., at the same hierarchical level), in a “null” (unanswered) state, that is yellow because its “Tumor Description” QO is optional.
6. Three sibling AOs that are children of the #5 AO that are greyed-out (in a “disabled” state) because their parent AO (#5) has not been selected.
7. An AO that is a sibling of three others above it with another AO’s children (#6) in between.
8. An AO of a required QO that is red because an answer is required, but no AO is selected (all in a null state). If answered, the FF would turn all three green.
9. A command button object (CO) that shows the status of a form’s section. It also enables the user to display (jump to) a section if its CO is clicked. It is colored yellow because the section has only optional questions.
10. A section CO whose green color indicates that required questions that have all been answered.
11. A section CO whose red color indicates that at least one required question has not been answered.
12. A CO that enables the user to save a form and/or send the results to remote locations.
13. A CO that creates a human readable report that shows the sections, questions, and inputs for answers given.

C) Advantages of our SDC Form Filler

Our SSF-based FF capabilities exceed SDC’s current web-based implementations that convert an FDF’s XML to HTML with XSLT [35]. Only our FF: 1) executes complex inferred skip logic routines based on each form object’s type, hierarchical relationships, and current state; 2) provides visual cues that simplify data inputs and outputs, that includes color-coding to indicate requirements and form completion status, as well as enabling and disabling objects based on hierarchical relationships; 3) converts the FDF XML into a hierarchical spreadsheet table that provides a simple computer-readable format and clear human-readable view of all form control attributes (properties) and captured data; 4) uses its built-in computational capabilities to implement any mathematical and logic-based rules from the desktop using local resources; 5) transforms data formats for optimal import into any store; 6) loads blank and populated forms anytime without connecting to the Internet for maximum portability; and 7) can transmit data and FDFs via any network architecture.
VII. SSF HIS SUPPORT FOR WPIC

The SSF can be used to develop applications for any situation in which there is need for a low-cost, convenient, secure, flexible, and resilient information and communication technology that is available without continuous network connectivity. In addition to helping improve EHR functionality [36], SSF apps are well-suited to support the WPIC strategy.

The WPIC strategy focuses on the development, use, exchange, and evolution analytic models that are built collaboratively and shared globally to emerge knowledgeable insights able to transform healthcare into a system that (as shown in Fig. 4):

- Integrates all of healthcare by incorporating: 1) Well-care, which focuses on preventing physical, mental and mind-body health problems from occurring or worsening through healthy living, wise decision-making, responsible action to deal with distressing life situations, and using effectively coping strategies with 2) Sick-care, which focuses on treating physical, mental, and mind-body health problems [37].
- Addresses the needs of the whole person, i.e., one's physical health (body), mental/psychological health (mind), and the mind-body connection (“holistic“ health). In other words, it views an individual as a whole entity, whose body and mind are interconnected.

WPIC’s goal is to improve people's health and wellbeing by helping them reduce their stress & distress levels (i.e., increasing peace of mind) and change their maladaptive behaviors (i.e., making lifestyle and treatment compliance changes) via computerized evidence-based decision support, collaborative knowledge transfer/sharing, coordinated action, patient counseling, and education [38].

If these needs are satisfied, it is likely to spur dramatic improvements in the quality, safety, and affordability of healthcare and preventive services across the globe. This would bring greater value to healthcare patients [39], [40], [41].

In technology limited environments (such as third world countries), the low cost and efficiencies of SSF apps create a “low bar to entry” by minimizing economic barriers that would otherwise hinder their participation in WPIC-focused networks.

The needs above can be satisfied by enabling EHRs (or other tools) to share patient information with all relevant parties to build the kind of research-based knowledge required to support clinical and financial decisions that promote continuous improvements in care quality, efficiency, and safety.

One way this can be done is through use of the SSF to develop and implement novel EHR companion (“bolt-on”) and stand-alone applications—such as the SDC form filler and other SSF apps and prototypes we have developed—which would support the WPIC model mentioned above.

Other SSF apps we have developed to support the WPIC strategy include: 1) a comprehensive, whole-person, personal health encyclopedia that provides self-management and problem solving support that integrates patient-generated [42] with clinician-obtained data; 2) a simple, sensible, and safe way for care teams to improve key processes supported by essential information about patient care and business operations through referral management, clinical messaging, and care coordination; 3) a clinical pathways tool that provides care decision support and outcomes analytics for process improvement; 4) a tool that visually depicts associations between a person’s health status, treatments, and significant life events in a timeline that is easy to understand; 5) query, analytics, advanced visualization, and collaboration tools that promote widespread data access and knowledge-growth; and 6) clinical business intelligence tools that use financial and operational data that provide insights for more informed decisions about how to make improvements in overall efficiency and effectiveness [43], [44].

VIII. OPEN SOURCE COLLABORATION

We share the IEEE GHTC commitment to support the development, deployment, and adaptation of technology focused on the achievement of humanitarian goals and sustainable development throughout the world.

As such, we intend to offer our SSF under an open source license that supports community collaboration, without inhibiting the potential for commercial use, to: 1) develop other SSF versions that use other spreadsheet programs, such as OpenOffice Calc and LibreOffice Calc, while we continue to evolve current Excel-based versions; and 2) build web-based apps that leverage the SSF HIS apps’ algorithms and processes. We believe such community involvement would enable our technology to have the greatest possible positive impact on human health and wellbeing worldwide. We welcome interested clinicians, developers, and other parties to contact us.

IX. APPENDIX

The appendix presents detailed SSF technical information.

Each SSF app uses the macros and worksheets contained in workbooks to structure and render data. Fig. 5 depicts the current structural components of the SSF app architecture; note: other components can be added as necessary.

---

**Figure 4**

[Image of a diagram showing the integration of physical health, sick care, well care, and holistic health]

**Figure 5**

[Image of a diagram showing the SSF Application Architecture]
A) Interacting SSF Workbook Components

1) The Data Transform Workbook (DTW) provides components and processes necessary for transformation of data from one format to another.

2) The Transport Method Workbook (TMW) provides components and processes necessary for sending and receiving data and information.

3) The Application Model Workbook (AMW) provides the main components and processes necessary to launch and run an application. It displays the primary UI to provide user control and has two built-in templates described below:

   a) Data Acquisition, Transformation, Transmission, Analysis, and Storage Template (DATAS): The DATAS consists of three modules (worksheets plus macros) and can utilize the components and functions of the DTW and TMW workbooks. The DATAS modules acquire, transform, integrate, analyze, and store data; the DTW and TMW provide the data transport and transform functions. The data can be acquired from user forms, databases, and delimited text files; purpose-built middleware can extend data access capabilities.

   The DATAS then stores the data in a DF, which can contain up to 17 billion pieces of data and metadata in a small file. For example, when stored in CSV format and zip compressed, the size of a DF that contains numbers in 10 million of its cells is under 60 KB in size; if it contains 10-character text strings in those cells, it is still only 700 KBs. This small file size saves storage space and enables huge amounts of data to be transported very efficiently.

   A DF’s contents can be organized for efficient rendering by arranging its data in arrays, lists, or in any other configuration depending on how the data are to be used. For example, if an application uses a portion of the data to render a chart, it is more efficient to organize the requisite data set as an array. If, instead, it uses certain data to populate a user form, it could be more efficient to arrange those data as linked lists.

   When an SSF apps stores analysis results in a DF, it enables rapid report generation because the “heavy lifting” (i.e., time-consuming data queries, aggregation, presentation-ready organization, and analytics) is already done, so the results are ready for the DFP to present immediately.

   DF1 is created by the AMW’s DATAS, which is stored it locally and transported it to AMW2 via the TMW (as indicated by the solid arrows).

   DF2 is created by AMW1’s DATAS (not shown) and is sent to AMW1 where its: 1) TMW retrieves the DF; 2) DTW performs any required transformations that may include use of spreadsheet-based maps; and 3) DF formats and presents the DF’s transformed contents (as indicated by the dashed arrows).

   b) Data Format & Presentation Template (DFP): The DFP is the second template. It is called (invoked) by the AMW as per Fig. 6. The DFP renders the contents of the DF it receives by applying formatting instructions to the DF’s contents, and presenting (displaying) the data in static or user-interactive tables, lists, charts, structured text, and advanced graphic visualizations. The DFP may also display the data in user forms. Furthermore, a DFP can be designed to combine the contents of different DFs to create composite reports.

Fig. 6 is an example of a network of nodes that use Excel and Outlook to exchange data. The first node retrieves data from a content source (#1) and transforms them into a DF that it sends to a series of other nodes (#2-6). Each recipient node extracts and uses what it needs from DF, may modify the DF as authorized, and returns it to the first node (via #3 & 6).

In these node networks, a sending node uses its DATAS functions to produce DFs, attaches the DFs to an e-mail message, and addresses the e-mail to receiving nodes. Each receiving node, in turn, retrieves the e-mail, decrypts and uses the attachment, and then formats and displays the DF’s content via its DFP. Sending nodes can: 1) transmit their DFs to a centralized node that automatically adds the DF’s contents to a database for aggregation and analysis or 2) use mesh node-to-node networks (a federated/distributed structure) in which many nodes can send and receive DFs directly between them.

![Figure 6](https://example.com/figure6.png)

X. REFERENCES


