

Electronic Procedures for Medical Operations in Space

Mary Beth Hudson¹
TRAC Labs, Webster, TX, 77598

Arthur Molin.²
S&K Aerospace, Houston, TX, 77062

and

David Kortenkamp³
TRAC Labs, Webster, TX, 77598

Electronic medical procedures are being used aboard the International Space Station. These XML-based procedures provide the ability to link in multi-modal information that offers the astronaut more information than the original paper counter-parts. Writing XML-based procedures is difficult using traditional word processing systems. Instead, an easy to use drag and drop editor, called PrIDE, was developed so that physicians themselves, rather than procedure writers, are able to develop procedures more quickly and consistently than has been done in the past. We are now realizing many of the other benefits of these XML-based procedures.

I. Introduction

ASTRONAUTS currently use the International Procedure Viewer (IPV) to view operational and medical procedures while in space. Medical procedures were the most recent addition to the collection of documents viewable by IPV and we are just now taking advantage of the benefits of modeling medical procedures in IPV's XML format. Prior to using XML, medical procedures were written in Microsoft Word. But due to a redesign of the medical kits that were to be used in space, many of the medical procedures needed major overhauls. This led to the desire to rewrite them so they could be viewable using IPV. However, writing procedures directly in IPV's XML format is not feasible. For operational procedures, there were a limited number of experts who used a licensed commercial tool and their knowledge of XML to convert Word documents to IPV's XML format. The procedure author would create the initial procedure in Word, then the XML expert would translate that to XML. After making the translation, they would return the IPV documents to the procedure author and they would go through one or more rounds of negotiations to make the procedure look as close to the Word document as possible.

When writing the medical procedures, the authors are expected to follow the Operations Data File (ODF) standards. The ODF is a one-hundred-and-fifty-page document that specifies all of the formatting standards for NASA documents that are to be used aboard the International Space Station (ISS). It includes details such as how far to tab over specific items, when to italicize, bold or underline, and a large list of symbols that take the place of words in commanding sequences. The procedure author bears the burden of ensuring that the document conforms to this large volume of standards. This takes a significant amount of time. And since Word is a free style word processing system and the authors are human, some medical procedures conformed to these standards better than others.

Another issue that came up was in the maintenance of procedures. It was common for a medical kit to be removed or updated for a specific flight. When this would happen, it would affect all of the procedures that called out a piece of hardware from the affected medical kit. It was a manual process to respond to this. The authors of the various procedures would be notified of the update, and they would have to remember which procedures that might be affected. It was a similar situation when a procedure was deleted or renamed. The process relied on the memories

¹ Senior Scientist, TRAC Labs, Inc., Webster TX 77598.

² Program Manager, S&K Aerospace LLC, Houston, TX 77058. AIAA Member.

³ Senior Scientist, TRAC Labs Inc., Webster TX 77598, AIAA Member.

of the procedure writers and perhaps the aid of a string search. Given the changing responsibilities for who was managing which procedures and the hundreds of procedures that were accumulating, this was becoming increasingly difficult.

We proposed a solution to these issues that would allow the procedure authors, themselves, to create the document directly in the IPV format. Furthermore, the document would conform to the ODF standards without the procedure author even having to be aware of the standards. And we would simplify the maintenance nightmare. The solution involved using a newly developed drag-and-drop editor called the Procedure Integrated Development Environment (PrIDE)³, an XML schema for procedures called Procedure Representation Language (PRL)^{4,5,6}, a database of medical hardware, and some XSLT translators. PrIDE and its underlying PRL are used to create the procedures, providing a more structured document than their free-form Word counterparts. The database helps manage medical kits and the hardware so procedure maintenance is more tractable. The database can be used to ensure the validity of hardware called out in procedures on a per mission basis, so that procedures do not call for a piece of hardware that is not onboard. The translators allow the documents to be converted to different formats so other viewers, such as IPV, can display them.

Our goals were to: 1) remove the human error caused by the interpretation of ODF standards, 2) ensure correct and consistent nomenclature for the medical equipment by linking to a hardware database, 3) provide a better search utility for hardware and procedure references than a string match, and 4) make sure the procedures are interoperable between PrIDE and IPV. We also incorporated new editing tools specifically designed for the medical procedure writers that made their jobs faster and easier (such as adding the ability to drag-and-drop a vital signs table, creating a personal palette, and importing Excel™ tables and IPV procedures).

II. Approach

To achieve our goals we took advantage of existing work we have previously done in authoring operational procedures. The PrIDE editor and the PRL schema were formerly developed for operational procedures. We updated them to reflect the new drag-and-drop commands that are used by the medical community. For example, a “Vital Signs Table” command was added that would create a table whose rows and columns were already labeled with typical vital signs and the frequency of the readings. We loaded medical equipment information into a drop-down list organized as medical kits, so users can drag a piece of hardware from a medical kit into the procedure. It appears as a string of text that cannot be edited (the text is treated as an object), but underneath XML stores information about the hardware, such as the unique id, a location, and the name of the medical kit it belongs in. This helps when trying to determine which procedures are affected when a medical kit is modified. We wrote XSLT translators that would convert the PrIDE XML format to IPV’s XML format and vice-versa. Figure 1 shows an overview of this architecture.

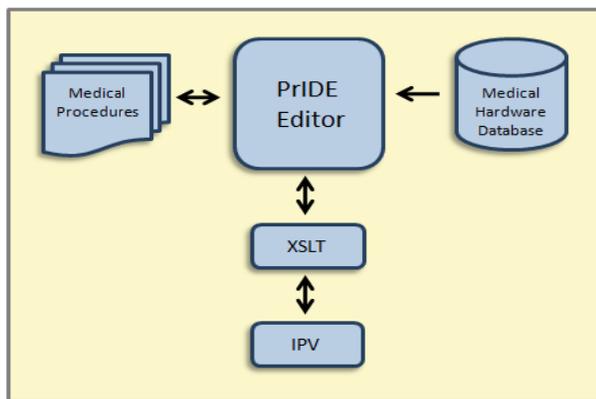


Figure 1. Architecture of the system used for creating medical procedures. The PrIDE editor is used to create medical procedures. It gets equipment information from a medical hardware database. The procedures it creates are interoperable with the ISS procedure viewer, IPV.

A. PRL - A Language for Modeling Procedures

PRL is an XML schema that underlies the PrIDE Editor. It defines the set of allowable tags that can be used to describe a procedure. In PRL, the highest level is a procedure tag that marks the beginning of a new procedure. Each procedure consists of steps that describe smaller tasks within the procedure. Steps themselves can have blocks that are containers for organizing a group of instructions that are associated in some manner.

1. Procedures

A procedure is the top-level entity in PRL. Each procedure has a human-entered name and number. Each procedure also has a unique identifier. A procedure can contain a block of “meta-data” with information about the procedure such as the author, comments, revisions, etc. A procedure can also contain an objective and informational items, such as Notes and Warnings, or a Parts List.

2. Steps

A step has a specific purpose or goal within the procedure. Each step has a human-entered name, a number that is generated sequentially and a unique identifier. Each step has an optional information statement, which is human-readable text that can provide additional information to a human performing the step.

3. Blocks

Blocks are wrappers that contain the instructions necessary to accomplish the step. The most basic block is an Ordered Block, which contains one or more instructions that are executed one after the other. An Unordered Block contains one or more instructions that can be executed in any order. Other block constructs offer control over execution flow such as if-then, repeat-until and while. Each block can contain other blocks or instructions.

4. Instructions

Instructions are the atomic actions of PRL. There are a wide variety of instructions, often tailored for different disciplines. The most common instruction in medical procedures is a Manual Instruction. A Manual Instruction is free-form text that describes an action a person must take. Other more structured instructions are: a Record Instruction, which records a reading, a Physical Device Instruction, which sets a device to a specified state, a Verify Instruction, which verifies a device is set to a specified value, and a Call Procedure Instruction, which references another procedure to be performed as part of this one. There are also informational instructions – those that provide additional information during the procedure, but do not require any action to be taken. These include: a Note, Caution, or Warning statement, a Table, a Figure, or a Hyperlink (which links to other files, such as audio or video files so additional information can pop up in a neighboring window while performing the procedure). It is also possible to include automation instructions that interact directly with other systems electronically once a step is reached, but the medical operations procedures do not use this feature.

B. PrIDE - An Editor for Creating Procedures

PrIDE is an Eclipse-based, drag-and-drop editing environment to create procedures based on the PRL Schema. The PrIDE User Interface has pull-down menus with icons for each step, block, and instruction. Figure 2 shows the

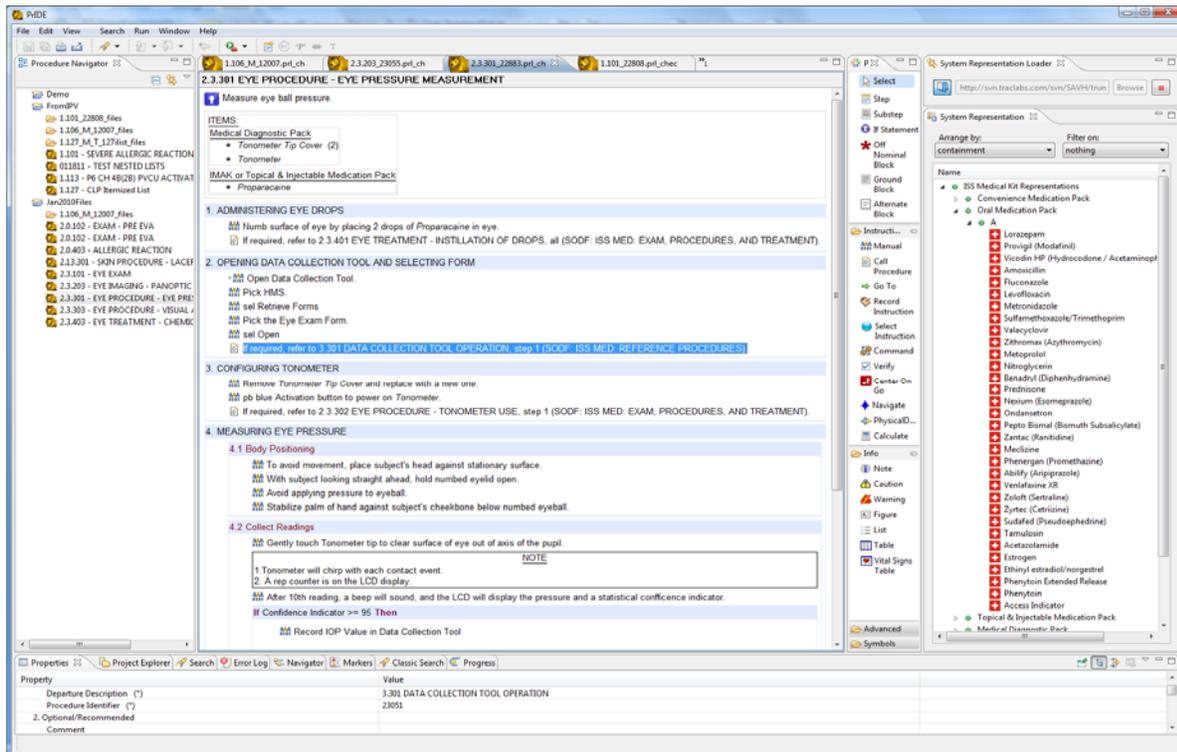


Figure 2. The PrIDE User Interface showing an example of a medical procedure. The leftmost box lists the names of procedures. The large box to the right of this is the procedure being edited. The narrow box to the right of the procedure is a palette of operations that can be dragged and dropped into the procedure. And the rightmost box is the list of available hardware, organized in medical kits, from the hardware database.

PrIDE User Interface. The Procedure Navigator window on the left side lists the procedures that have already been created. The large window to the right of the Procedure Navigator is the place where the procedure is created and edited. The small window to the right of that is the pull down menu of empty steps, instructions and other supporting information types. The user drags empty steps and instructions from this window into the editor. The window to the right of that, with all of the red crosses, is the database of medical items from which the user can select hardware to include in instructions and lists. This ensures that the nomenclature for the hardware is consistent throughout all of the procedures. In fact, the nomenclature to be used in procedures is one of the fields defined in the database. The procedure also stores additional fields from the database in the underlying XML. For example, it stores the unique ID, the name of the medical kit that the hardware belongs in, and the location of the hardware. All of this information is in the underlying XML hardware representation, but the user only sees the name of the piece of hardware when it is dragged into an instruction. What is displayed in the procedure may be context dependent. For example, at one point, we automatically added the name of the medical kit when a piece of hardware was dragged to a Parts List. This allowed the astronaut to locate the med kit that the hardware belongs in by looking at the Parts List. When searching the procedures for a piece of hardware, PrIDE searches using the hardware ID from the database, rather than relying on a text search which traditionally only works when the items are spelled out exactly the same.

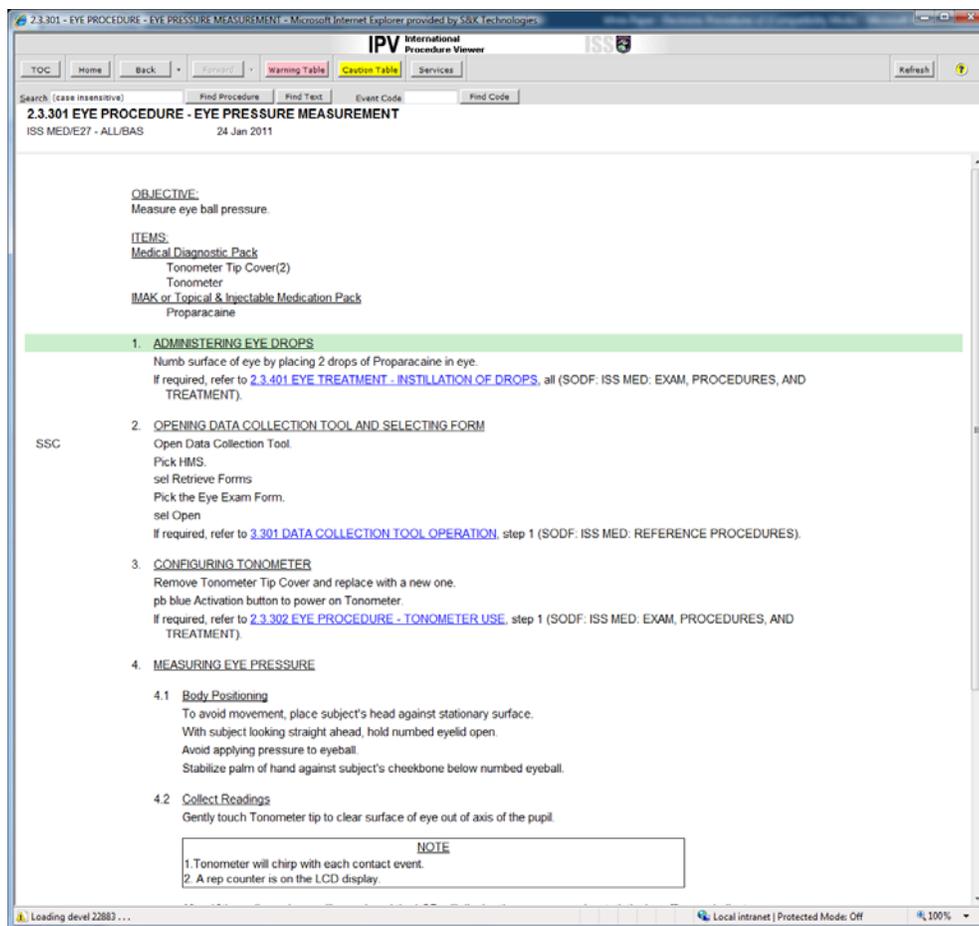


Figure 3. View of the same medical procedure in IPV. Figures 3 and 4 depict the same medical procedure. The content is identical, but the format is different depending on which platform the procedure is displayed.

To create a procedure, the user selects File> New Procedure and then answers a few questions about the procedure, such as the name of the procedure, the unique ID assigned to the procedure, the author's name, the date, the book the procedure belongs to, etc. Most of these fields already have the default values filled in. Then a new procedure with only one empty step populates the editor. The user can name the step and start dragging in more steps and instructions. The available steps and instructions are displayed as icons in the palette to the right of the procedure. As instructions are dragged in the user fills in the text associated with that instruction. For example,

many procedures start with a parts list. So the user would drag the list icon onto the page and enter the title of the list where prompted. Then the user would type in the items that belonged to the parts list (without having to worry about what is capitalized, what is underlined, and what is indented). Similarly, the user would drag step and instruction icons from the palette to the procedure and then type in the text associated with that step or instruction. The user continues this process of dragging items and further specifying them until the procedure is complete.

Once the user has dragged all of the items into the procedure, the items can be copied, pasted, and rearranged until the procedure contains exactly the information in the order that the user wants it to appear. Finally, when the content is complete, the document will be translated into its final format. The user performs this translation by selecting “Translate to” and selecting the desired format (such as “IPV”). To do the translation we use XSLT (Extensible Stylesheet Language Transformation). This style of development allows the procedure developers to write procedures without having to worry about the final format. If new formats become relevant in the future, it is a simple matter to add another XSLT translator to the PrIDE interface.

Figures 2 and 3 depict the same procedure. Figure 2 shows it under development in the PrIDE editor and Figure 3 shows it in its final format in the IPV Viewer. While the content is the same, they both have a different look to them. This demonstrates one strength of an XML-based system – the data is separate from the format. The data can be displayed differently on the different platforms. The format for displaying the data is captured in a style sheet for each platform. The same procedure can be displayed on many different devices, such as a full screen terminal or a hand-held device, without having to rewrite the procedure.

C. Database Connection

PrIDE provides a direct connection to a database specific to the procedures that are being developed. For the medical operations procedures, this database contains about 260 items divided into nine medical kits. Each medical kit has an “A side” and “B side” which tell the astronaut where in the medical kit to find an item. A medical hardware database called HMS Inventory Tracking Tool (HITT) was developed to manage the medical hardware specific to the Health Maintenance System (HMS) procedures. Other generic pieces of hardware, such as batteries and other pieces of medical equipment not specific to the HMS may be found in other databases. PrIDE uses the information only from the HITT database to help the users write their procedures. The HITT database contains information about each piece of medical hardware, such as the agreed upon name or nomenclature that should appear in procedures, a unique ID, its location in the med kit, and a description (often an alternate or generic name for a drug). All of this information is stored in the PRL file, even though only the nomenclature is displayed in the procedure—the remainder of the information is available for other processing of the procedure.

One of the issues that the procedure developers face is that the procedures are required to be written before the contents of the medical kits are finalized. But the procedures call out hardware from the medical kits. As a consequence the procedures have to be regularly updated as new versions of the medical kits are released. One of the major advantages to our approach is that the existing procedures can be easily compared to new releases of the medical kit database—elements that have been changed or removed from the database can be flagged for update in the procedure. We discussed the possibility of automatically updating the procedures when the medical database changed, but the procedure authors preferred not to cede control to the software in this matter—they liked the ability to review every change before it was made. Nonetheless, this capability was singled out as one of the biggest contributions to the project—the procedure authors felt that they would not have been able to produce as complete and high quality a product using Word because of the impact of medical kit changes after they started writing procedures.

D. Translation

We used XSLT to translate between PrIDE’s underlying XML format and IPV’s underlying XML format. To test the translation, we wrote a translator from IPV back to PrIDE which turned out to be serendipitous because later it was used to import existing IPV documents into PrIDE. We were not able to model all constructs that IPV models due to resource limitations. However, we did support every construct that medical operations required at the time. Furthermore, PrIDE had some features that IPV did not support so some translations could be lossy. XSLT simply does not translate things it does not recognize.

The translation to IPV from within PrIDE allowed the user to see what the document looked like on IPV as it was being developed in PrIDE. This was a crucial step in providing the user with confidence to use PrIDE since the final document was delivered in the IPV format. The bi-directional translation allowed the users to pull existing IPV procedures used in operations into PrIDE so they could see how certain commands were used. They could also copy parts of existing IPV procedures into their medical procedures.

We were very happy with the translation from IPV to PrIDE, particularly on operational procedures it had never seen before. It managed to correctly translate somewhere between seventy to ninety percent of random IPV operational procedures. The missing parts were the unsupported constructs, so it would be a matter of adding the constructs to PRL and PrIDE to get a complete procedure.

We discussed the possibility of translating our procedures to another XML-system developed for viewing medical procedures on a handheld device. GuideView² is Graphical User Interface display for medical procedures that guides the user through the procedures and is targeted towards handheld devices, such as Blackberries, Palm Pilots, ITouch etc. GuideView has its own proprietary procedure representation in XML that has links to audio, video, and other external files so it can provide additional instruction during the execution of the procedure. In theory, if we had made the translation, then all of the existing IPV procedures could be immediately made available to view on GuideView, simply by translating the IPV to PrIDE and then the PrIDE to GuideView. This shows another strength of XML-based documents – once you have a two-way translation from one system’s XML to another, then both of the systems can view all procedures originated by either system. If someone adds the ability to translate their system to one of the existing systems, then all systems get the benefit of being able to view the new system’s documents. In effect, each new system/translator adds to the pool of documents that will be available to all systems and the new system gets immediate access to this large pool of existing documents.

III. Operations

As soon as the initial prototype was delivered in June 2010, a physician at NASA-JSC started using it to develop the ISS Medical Checklist, a book that contains over one hundred medical procedures. Figure 4 shows an example of a medical procedure he developed using PrIDE. This procedure shows what to do in case of a severe allergic reaction

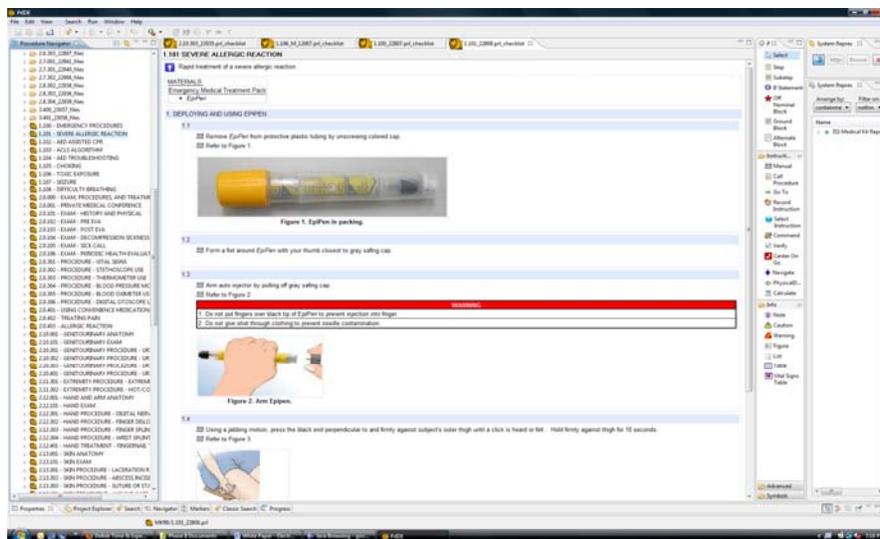


Figure 4. Example of a severe allergic reaction medical procedure being developed in PrIDE.

An example of a medical procedure, developed in PrIDE, that includes text, images, warning boxes.

reaction. Besides the steps and instructions, it includes components such as a parts or materials list, some pictures, and a warning box. It is easy to see the hierarchical nature of procedure. For example, Step 1 is a broken down into substeps (1.1, 1.2, 1.3 and 1.4) which have one or more instructions, warnings, or images. Figure 5 shows the same procedure after it has been translated into IPV and then Microsoft Word. Note that the physician never had to specify any of the spacing, numbering, indenting, capitalizing, or bold or underlined text as seen in the Microsoft Word document - this was handled by the software, following ODF’s well-defined formatting standards.

During the development of the book, the physician provided us with feedback that not only helped ensure that all components of the medical operations procedures could be modeled, but it helped us to determine which functionalities would have priority. Because medical procedures have instructions that are common among more than one procedure, we added a “Personal Palette”. The Personal Palette is used when the same set of instructions are used in several procedures. The user can select the portion of the medical procedure that is commonly used and copy them to a buffer. This buffer is given a name by the procedure author, for example, “Blood Pressure Instructions”. From then on, when the author wants to place those instructions into a procedure, he pulls down his

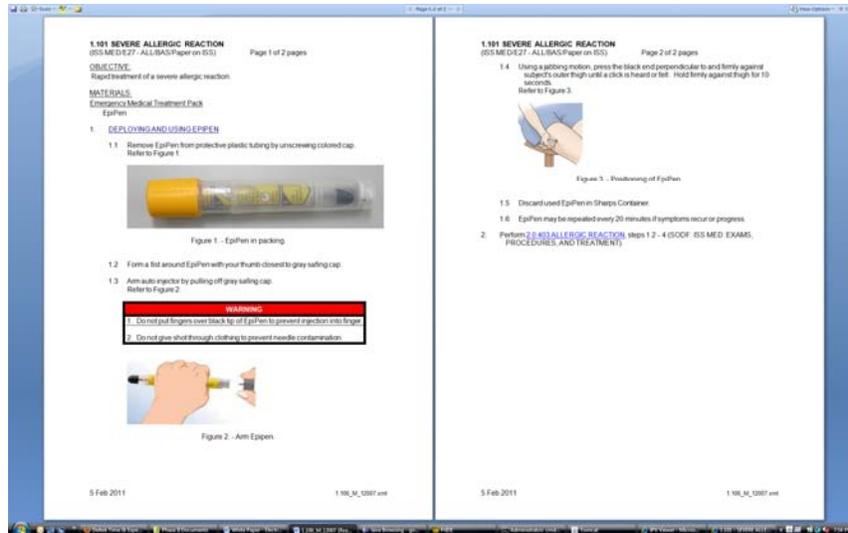


Figure 5. The same severe allergic reaction medical procedure in Microsoft Word. *This medical procedure that was developed in PrIDE, translated to IPV, and then translated to Microsoft Word*

Personal Palette and drags the “Blood Pressure Instructions” into the procedure. A copy is made where the cursor is clicked. Theoretically, this could be made into a separate procedure which the author would just call out – Call Procedure 1.2.77; Taking Blood Pressure – however, there are many cases when the subset of instructions do not warrant their own procedure.

A specific request made was to import existing Excel tables. A table is a component that can be dragged into the procedure from the palette. Then the headings, rows, and columns are filled in. It turns out that there are existing Excel spreadsheets containing large amounts of data that the procedure author did not want to retype. So we added the ability to import these items directly into the procedure, much the same as if it were dragged in off the palette.

When making a Call Procedure instruction, the user no longer has to type in the name, procedure id, step numbers, etc. Instead, the user can find the procedure in his Navigator Window and drag it into the editor. This will create an instruction that Calls the procedure dragged in with default steps “all”. From there the user can edit the step numbers to be performed if necessary.

A Vital Signs Table was added as an instruction. Once dragged in, it also can be modified. The number or frequency of readings can be changed as well as the vital signs that need to be taken.

Lastly, a new search utility was added. To find a piece of hardware or a reference to an existing procedure, the user can just right click on the item to be found and then all of the procedures in that folder are searched over. (The searches work by IDs since these are not strings, but rather objects.) The results are displayed in a window at the bottom of the display. The name of each procedure that has a found

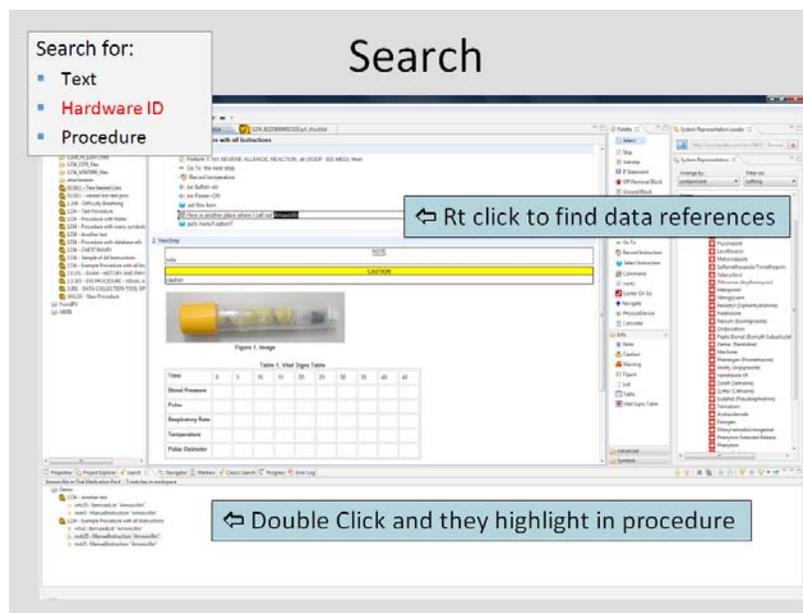


Figure 6. PrIDE’s Search Tool. PrIDE can do a string search, but uses IDs when searching for pieces of hardware or references to other procedures. Right clicking on an item in the procedure or medical kit will instigate a search on that item. Clicking on the results of the search brings up the procedure with the item highlighted.

reference is listed followed by each instance in that procedure that was found. The procedure names may be expanded or collapsed to show or hide the instances. This gives a global view of which procedures contain a piece of hardware as well as a view of how many times that hardware is called out in a given procedure. To go to the place in the procedure where the hardware is called out, you just double click on the instance and the procedure opens in a new window with the instances highlighted. The same procedure works when searching for a text string. This greatly helped solve the issue in finding out which procedures were affected when a medical kit was changed out. Figure 6 shows the interaction with the search tool.

To simplify the user interface, we removed several PRL features that were used in the past by other groups, but that were not needed by medical operations. These features support automated execution of procedures. We are developing software that will allow computers to read PRL and send procedure commands and verify telemetry automatically¹. This capability is desired for routine, operational procedures. Most medical procedures are not routine and do not have electronic commanding capability. Thus, automation is not a primary concern. However, there may be some future benefits. For example, when using medical devices the readings from them could be automatically inserted into the procedure at execution time. This would save the user from having to record the readings himself and may also prevent human error in the readings.

IV. Results

Originally the physician who wrote the ISS Medical Checklist was using PrIDE to help test out its feasibility and give us feedback on its development. In the end, however, he used it to create the majority of the procedures for his book. (PrIDE was unable to handle some non-standard procedures such as flow charts.) We delivered the first prototype to him in June 2010 and he published his book in February of 2011. It took him only eight months to complete a book of about 120 medical procedures from start to finish. He told us that it was quick and easy to create procedures using PrIDE, and the turn around time on modifications was much quicker than it traditionally had been in the past. Part of this may be due to the fact that the step of passing the procedure to the XML editor for updates each time was omitted since the procedure author can now make his own updates. Another factor may be because the one physician wrote the whole book and so he was intimately familiar with all of the procedures. He said he was also able to apply generic edit requests to multiple procedures at the same time so he could respond faster to the reviewers' requests. He mentioned that the procedures were more consistent across the entire book. Again, this could be due to the fact that he wrote the entire book, but it could also be due to the fact that formatting standards were now automatic. In some final interviews with him he noted that

Using PRIDE, Medical Operations was able to create a polished product in a short period of time. It is doubtful that this would be accomplished with WORD. Although not part of the standard procedure authoring process, PRIDE's drag and drop functionality, which allows for the rapid creation and revision of SODF compliant procedures directly into XML, promptly gave it a key role in the build of the new book. Because PRIDE output can be immediately uploaded into and viewed in IPV, design concepts and draft procedures could be instantly tested and revised. In addition, PRIDE's whole book search and editing functions enabled editorial control which resulted in multiple, refined versions of the book being quickly produced (even in our compressed schedule) before having to go to print. This resulted in a more consistent and much improved product than would have otherwise been made.

It has been about seven months since the book was published. To this day, the physician continues to use the PrIDE prototype that we delivered to him. He wishes that the revision process were more integrated into PrIDE, because for small changes, it is easier to tell the SODF folks what to update in the procedures than to update the document using PrIDE and then highlight the differences between the old and new versions. For new or documents with major changes, he still relies on PrIDE.

V. Conclusion

Our goal was to provide a prototype system on which to develop XML-based medical procedures that would allow the procedure developer to focus on the content of the procedure rather than spend time formatting to the lengthy and strict ODF standards. We have successfully delivered the prototype, which has been used to develop over one hundred medical procedures for ISS operations. The procedures created are more consistent in format and content than their predecessors. Today, these medical procedures are being used by the astronauts aboard the ISS.

The prototype is also a research tool for creating and using medical procedures electronically and it can be used for further investigation and incorporating of other medical informatics technologies. The software is available to any government organization through a licensing agreement.

Acknowledgments

M.B. Hudson would like to thank Dr. Stevan Gilmore for the tremendous support and feedback he gave us in deploying PrIDE for Medical Operations. Without him, PrIDE for Medical Operations would have remained a good idea instead of an operational reality. Authors M.B. Hudson and A. Molin would also like to thank Dr. Kathy Johnson-Throop for the direction and contacts she provided so that PrIDE was able to be connected with the rest of NASA's operations. This project was funded by a NASA Small Business Innovative Research (SBIR) contract NNX09CA96C and the authors thank Dr. Jeremy Frank of NASA Ames Research Center and Lui Wang of NASA Johnson Space Center for their support with this contract.

References

- ¹Bell, S., and David Kortenkamp, "Embedding Procedure Assistance into Mission Control Tools" *Proceedings of the International Joint Conference on Artificial Intelligence Workshop on AI in Space*, 2011.
- ²Iyengar, M.S., Sarkar, S., Bacal, K., Defouw, G., McCulley, P., and Hurst, V., "GuideView: Structured Multi-modal Delivery of Clinical Guidelines" *American Medical Informatics Association Annual Symposium Proceedings* [online journal], Vol. 2005, URL: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1560583> [cited 29 Aug 2011].
- ³Izygon, M., David Kortenkamp, and Arthur Molin. "A procedure integrated development environment for future spacecraft and habitats," *Proceedings of the Space Technology and Applications International Forum (STAIF 2008)*, available as American Institute of Physics Conference Proceedings Volume 969, 2008.
- ⁴Kortenkamp, D., R. Peter Bonasso, and Debra Schreckenghost. "Developing and executing goal-based, adjustably autonomous procedures," *Proceedings AIAA InfoSys@Aerospace Conference*, 2007.
- ⁵Kortenkamp, D., R. Peter Bonasso, and Debra Schreckenghost. "A procedure representation language for human spaceflight operations," *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS-08)*, 2008.
- ⁶Schreckenghost, D., Tam Ngo, Robert Burrige, Lui Wang, and Michel Izygon. "Remote task-level commanding of centaur over time delay," *Proceedings of the Space Technology and Applications International Forum (STAIF) (AIP Conference Proceedings Volume 969)*, 2008.