# VISION OF THE VIRTUAL PROGRAMMER

## *Steps Towards Change in Instrument Systems for Implantable Medical Devices*

Touby Drew and Steve Goetz

*Neuromodulation, Medtronic, Inc., 800 53rd Avenue NE; MS N210, Minneapolis, USA*

Keywords: Implant, IMD, Instrument, Software, Trends, Architecture, Mainstream, Programmer.

Abstract: Active implantable medical devices increasingly depend on and interact with external systems of instrument hardware and software. Based on our work in defining and refining the direction of next generation instruments, we submit that there are and will increasingly be a trend towards complex, mainstream instrument systems, which are distributed, decoupled and part of rich modular information ecologies. As this shift occurs, important challenges arise and must be met with domain-specific solutions including those in the areas of security, repartitioning, and changes to instrument architecture and development.

## 1 INTRODUCTION

Traditionally instruments used to interrogate and configure implantable medical devices (IMDs) have tended to be isolated and proprietary. This paper discusses: (i) the trend for these instruments in the direction of mainstream computing as well as increasing complexity, connectivity, and loose coupling. (ii) security, obsolescence, and development issues becoming critical to the future of the IMD industry as a result; and (iii) means to address these challenges with system architecture, shared infrastructure, and changes in development.

## 2 BACKGROUND

Currently there is a range of active implantable medical devices (IMDs) and instruments which interact with them. This includes implantable pacemakers, cardioverter defibrillators (ICDs), neurostimulators (INSs), pumps, and other devices which are created by companies such as Medtronic, St. Jude Medical, and Boston Scientific. The scope and prevalence of use of IMDs is significant and growing as “the market for electronically driven implantable devices is set to expand significantly… and will reach $33.8 billion by 2009” (Groen, 2007).

“Instruments” or “programmers” are the systems which accomplish “non-invasive reversible alteration of the electronic performance of an implanted device” (Harthorne, 1983). “Programming” has come to mean any interaction with the data or settings of an IMD. Configuration of IMDs was first accomplished percutaneously, but by the 1980’s sophisticated, wireless communication “allowed for the possibility of speedier transmission, bidirectional telemetry, and greater proprietary/security access coding” (Schoenfeld, 1993). As programming evolved, instruments became complex, special purpose computing platforms incorporating communications technology for interfacing with the IMDs.

The term “telemetry” describes the parts and protocols of “programming” directly involved in communication between an IMD and an instrument. Telemetry is proprietary, varied, and optimized for domain constraints (e.g. ECG streaming). At Medtronic, for example, telemetry in use today varies in physical layer encoding, data layer manipulation, and in application messaging to the extent that no single instrument supports all of them.

Recently, telemetry tends towards longer range (e.g. Conexus™), standardization (e.g. MICS-band), and decoupling from other aspects of programming. This decoupling is seen in the proprietary-cable-connected 8840T telemetry head which has its own processor and software but is actually part of the handheld 8840 N’Vision™ Clinician’s Programmer used with Medtronic Neuromodulation products.

In addition to telemetry interfaces, programmers often have other specialized functionality. For example, many cardiac instruments include a pacing



Drew T. and Goetz S. (2008).
VISION OF THE VIRTUAL PROGRAMMER - Steps Towards Change in Instrument Systems for Implantable Medical Devices.
In *Proceedings of the First International Conference on Biomedical Electronics and Devices*, pages 156-159
DOI: 10.5220/0001050401560159

system analyzer and a strip chart printer for ECG waveform. These have historically been tightly integrated with telemetry and IMD configuration software on special purpose computing platforms to define traditional programmers.

# 3 INSTRUMENT TRENDS

Instruments have begun to incorporate modern, mainstream computing platforms, driven by the market and increasingly capable IMDs. Key aspects of this trend include growing complexity as well as more connectivity and looser coupling between instruments and their component technologies.

## 3.1 Complex and Polymorphic

IMDs have grown beyond their pacemaker roots – INSs alone are being applied by a number of companies in therapies ranging from gastroparesis to epilepsy. As the therapy, diagnostic, and other capabilities of IMDs become increasingly complex and varied so does their programming, instruments, and information management. In one division of Medtronic alone it was estimated that more than 150 applications accounting for several tens of millions of lines of software code are being used and maintained. Though some of this is due to the long support life of IMDs (and corresponding legacy in instruments), some is a result of change in technology and user desire. For example, telemedicine has driven the development of distributed instrument systems like the CareLink™. To this extent, a single instrument system is difficult to achieve and instrument solutions are tailored to user and business needs (e.g. of particular therapies).

## 3.2 Connected and Loosely Coupled

As new instrument capabilities are introduced, there is increasing pressure to modularize, distribute, and connect instrument functionality (across instruments as well as other external systems). One example is the integration of remote monitoring systems (such as CareLink™, Housecall Plus™, and LATITUDE™) and traditional instruments. A second driver is the increasing prevalence of personal therapy manager (PTM) instruments, which a patient can use to journal or adjust therapy in a manner different from, but complimentary to, clinicians' programming.

There are also internal business motivations for modularizing and connecting instrument functionality. As noted, it is essential that telemetry remain connected, but it is a common target for decoupling from the rest of an instrument. This is partly because of its unique constraints and proprietary nature, but also to facilitate the abstraction and reuse of hardware and software which are required by multiple instruments.

In addition to the growth of connectivity and reuse across programmers, there is increasing connection between programmers and existing clinical and general purpose infrastructure. For example, though proprietary printing capability is built into programmers such as the Medtronic 2090, the 8840 prints via IR and may support other standard interfaces in the future. Similarly, clinicians are increasingly demanding that instruments allow them to export information to and view data from electronic medical record (EMR) systems and their mainstream PCs, where increasingly standardized, networked systems are changing the face of care.

## 3.3 Instruments and Modern, Mainstream Computing

As previously described trends continue there is crosspollination to the extent that once-entrenched boundaries are now blurring. Companies like Mednet and Raytel allow clinicians to outsource IMD follow-ups. Patients can review and control their own therapy and physiology with patient oriented instruments, reports, and data systems. Clinicians are increasingly able to interact with patients, data, and IMDs, from their home to their patients', using mainstream tools and infrastructure.

In this way, instruments are using and becoming a small part of the rich, modern, general purpose computing technology pool. Starting with cautious steps into carefully controlled and customized commercial components and then connecting electronically for telemedicine and software distribution, programmers like the Medtronic 2090 have evolved and continue to grow in this direction. For example, Advanced Neuromodulation Systems (ANS) released a PDA-based instrument which prominently displays the HP logo on the front and a Microsoft Windows™ logo on the back (Rapid Programmer 3.0, 2007). Though there is some proprietary aspect of programmers like this, they are largely customizations built on general purpose computing platforms. This is apparent in the fact that the ANS Rapid Programmer™ was seen only three years earlier sporting a Compaq rather than HP logo (Pain Medicine Network, 2004).

Instrument manufacturers are realizing that in the near future isolation from the mainstream will be untenable as the race to meet growing needs reliant on complexity and connectivity progresses. We submit there is a double-edged ‘Red Queen’s race’ in that near continual evolution of not only proprietary instrument capability, but also its integration with mainstream technology will be important to remain competitive.

### 3.4 The Virtual Programmer Future

Design of next generation instruments should follow a direction consistent with the industry trends described previously. Some distance down this road we submit there may be a `virtual programmer’ that, unlike IMD instruments today, relies almost entirely on shared, general-purpose, existing infrastructure such as clinician’s PCs and a more virtual, distributed network of functionality. The programmer, in the form of hardware, operating systems, and other components provided by IMD manufacturers disappears – to be replaced by a changing, mainstream, connected pool of resources largely owned and maintained by others.

Commenting on this idea one of Medtronic’s regulatory experts noted that this would be a “revolutionary change rather than an evolutionary one” and could not be achieved overnight (Peterson, 2007). To this end, a first step might be to transition to the mainstream, buy and operate on a commercial off the shelf (COTS) platform, and separate from that any domain specific components which must remain proprietary. The IMD manufacturer would still control (in a regulatory and legal sense) the programmer, as has traditionally been the case. The second step would then be to transition to the use of general purpose, clinician owned and maintained PCs, operating systems and technologies in place of all but the essential proprietary components. To our knowledge no such clinician PC based instrument system exists today despite the potential hardware and support cost savings.

## 4 SHIFTING CONCERNS

As a result of our work towards a virtual programmer, we have begun to see what we believe are key challenges in security, obsolescence, and development which are and will be echoed across the industry. With simple, isolated instruments there is a low attack surface or room for misuse to motivate complicated security concerns. As trends continue toward complex, connected, mainstream components outside the control of the instrument manufacturer, associated migration and development costs beget significant additional concern.

### 4.1 Security

With instruments, there is the potential for some severe types of harm including discomfort, death, and exposure of protected health information (PHI). Complex, mainstream, connected, general purpose computing is notorious for having a high rate of breach due to a number of different vectors which are not of significant concern for traditional instruments. Windows operating systems (e.g. in the PainDoc™ programmer) are infamous for having their “survival time” measured in minutes when exposed to the Internet (SANS™, 2007). This combination begets a potential for high severity and high frequency security challenges. Also, in a connected, virtual programming environment, one can imagine new and frightening scenarios. One might no longer have to get within a few feet of each implant to program it or interrupt service to it and anyone with internet access could potentially access or affect a tremendous volume of PHI, patients, IMDs and instruments in patients’ homes and clinics. Instruments could become a risk to their host clinical environments and they could be severed from their increasingly important information ecology or inadvertently open a backdoor for hospital infrastructure to be compromised.

Though scenarios involving malicious attack against patients (e.g. Vice President Cheney via his ICD), addictive self medication, or even attempts to manipulate the stock of a public instrument manufacturer are regularly raised, perhaps the greatest risk lies in less specifically targeted security scenarios in which needed therapy is unavailable or delayed. In many cases instruments are essential to the therapy of patients to the extent that if programming is unavailable when changes are required or an IMD state is unknown there may be significant patient harm. The ability to program could be lost or interrupted by something as minor as the instrument software not having sufficient resources to operate (e.g. because malcode or user software has used up RAM).

### 4.2 Obsolescence and Development

Recently companies like St. Jude, and Medtronic (e.g. 2010 programmer) have begun basing their instruments on COTS platforms available from companies like Microsoft, Intel, HP and others from the broader computing market. With these mainstream components come mainstream capabilities, but also mainstream development cycles and obsolescence concerns.

Medtronic's 8832, for example, was based on the coupling of telemetry hardware with a Handspring™ Visor through the proprietary Springboard slot which is no longer be available. With the hardware platform effectively dead, this is an additional source of replacement cost and lost synergy relative to a more maintainable common PTM platform.

In the case of a virtual programmer the user and third parties would be rapidly altering their software environment creating numerous configurations and interactions which might cause issues. It would be unacceptable to allow users to keep their operating systems unpatched or to require IMD manufacturers to anticipate, support and test every permutation.

## 5 MEETING NEW CHALLENGES

Despite the magnitude of effort applied to topics related to IMD and instrument futures, we are unaware of any published work exposing these challenges or exploring paths to confront them within the domain and focus laid out earlier in this paper. This section attempts to summarize key approaches and examples related to architecture, infrastructure, and development changes which we submit are valuable to consider in this light.

Careful definition of the recently evolved concept of an independent, proprietary telemetry module (TM) exemplifies how careful repartitioning and organization of instrument functionality can be a valuable approach to consider. Separating critical and proprietary functionality into a decoupled component allows it to evolve and operate independently so long as a standard interface is maintained. Furthermore, an entire class of safety and security concerns related to the failure or unavailability of a COTS clinician's programmer could be addressed for example if the TM it used was developed to include independent "safe mode" and authentication capability.

As instruments become increasingly complex and mainstream, the infrastructure through which they are manufactured, operated, and deployed must fundamentally change from their expensive, isolated, monolithic, IMD manufacturer-proprietary past to leverage the benefits of the technical and business community they are joining. In the virtual programmer scenario, significant development, support and integration costs might be reduced or offloaded by leveraging networks, PCs, IT people, and other key resources from customers and groups which distribute costs. Companies like Intel and Motion Computing are already positioning for such roles in the vertical healthcare market with initiatives like Intel Health and the Motion C5.

Definition and use of standard, domain specific languages and tools may further allow for better (clear, constrained, abstract, etc) development path for programmers. These could include HL7, DICOM, those which may emerge from the IMD domain, and others (McDonald, 1997).

To this end, those who develop instruments may have to be increasingly open to two major changes: 1) extensive partnership, cohesive integration, and cross pollination blurring traditional product, business, and even industry/academic boundaries and 2) a phased approach with multiple steps to achieving long term goals through focussed evolutionary changes.

## 6 CONCLUSIONS

Instruments for programming IMDs form an interesting domain with trends towards decoupled architectures, connectivity, complexity, and the mainstream which are emerging as part of a poorly defined path towards solving evolving challenges in security, obsolescence, and development.

## REFERENCES

Groen, P. Implantable Medical Devices and EHR Systems. Last viewed June 25, 2007. http://www.hoise.com/vmw/07/articles/vmw/LV-VM-06-07-5.html

Harthorne JW. Programmable pacemakers; Technical features and clinical applications. In L Dreifus (ed.); *Pacemaker Therapy*. Philadelphia, PA, F.A. Davis, 1983. pp. 135-147.

McDonald C. The barriers to electronic medical record systems and how to overcome them. *J Am Med Inform Assoc*. 1997;4:213–21.

Pain Medicine Network. page10. *American Academy of Pain Medicine*: Vol. 19, No. 3, Summer 2004.

Peterson, C., K. Ruth-Jarmon, et al. The Programmerless Programmer. May 2007.

Rapid Programmer 3.0. http://www.ans-medical.com/smallwonder/ Last viewed June 24, 2007

SANS™. "Survivial Time." Last viewed June 24, 2007. http://isc.sans.org/survivaltime.htm

Schoenfeld, Mark H. 1993. *A Primer on Pacemaker Programmers Pacing and Clinical Electrophysiology* 16 (10), 2044–2052. doi:10.1111/j.1540-8159.1993.tb00998.x