

THE PANIO ORTHODONTIC SOFTWARE DEFINITION

ANTONIO MAGNI

ABSTRACT. PANIO is a standard to define an environment for orthodontic software development in an effort to solve the confusion present in currently available software. The project includes defining a free model for orthodontic software to adhere to as well as a basic, free open-source implementation of it. Once implemented, PANIO will be free, and all of its source open and freely available. Our intention is not to compete on the market, but to alleviate current developers from common bookkeeping tasks, while at the same time providing a very versatile solution for orthodontists.

1. INTRODUCTION

As physicians are relying more and more on computers, losing patient data could cause serious consequences. Developing a standard makes data portable, which means it can be opened and viewed by many different systems. This means that if a system fails, and the data is not stored in a proprietary format, one can still view the data with a different system, causing an increase in stability and portability. On top of these advantages, developers need only to implement a well defined standard when building their system, instead of having to design a new one. It is therefore understandable for many medical fields to develop their own standards (DICOM, a standard for image storage and transfer, Health Level 7, an organization defining medical reference information models (such as data models and, messaging systems), VISTA, a medical application platform).

Orthodontists keep complaining about the software available for their practices: they like them, but they don't seem to ever be satisfied with just one product. They like some feature from one software, another feature from a different package, and cannot use them all at once. The orthodontic software community does not yet have a standard developed, and as a consequence it's development grows wildly in each direction. In the next section, we take a closer look at the the specific problems of the orthodontic software community.

1.1. The problems. Programs are available, in many different varieties. But some orthodontists seem still to be unhappy. What is it they complain about?

- (1) *The orthodontist does not have time to purchase, learn, install and then maintain many different software packages and cannot limit him/herself to just one software vendor, because one that can provide a solution for all needs does not exist.*
- (2) *The orthodontist does not have complete freedom to purchase any software package available on the market: he/she is limited to the platform it is written for, because only few of the available orthodontic software packages provide a solution that is platform independent. Most of the time, it is the hardware already present in the practice that has the last word. This is inherently incorrect, as the user should be able to choose the platform based on the features it has to offer. For example, a user might like the new features offered in Windows XP, or the flexibility and stability of Linux but they cannot buy an IBM-PC compatible because [software package here] only runs on MacOS.*

(3) *Software developers keep spending energy rewriting functionality that is already available.* This means slowing down research and the development of cutting-edge technology: if an institution just recently came up with a new method for performing cephalometric analysis, and would like to make it available to the public, they should be able to do so quickly. Today, before they can even get to implementing their novel algorithm, they must first write a whole new program which contains the functionality already present in thousands of other software packages (as in patient record management). And in the end, nobody is going to want to run that program, simply because it means installing, learning and maintaining a new software package. This is scientifically counterproductive.

Is it really true that there is nothing available today that attends these three requirements?

1.2. Today's situation. Today there is a variety of software packages which assist the orthodontist in almost every step of her/his daily routine. Programs like *OrthoTrac*, *Orthoware* and *Advanced Ortho*[6] help manage patient data, *Dentofacial Showcase*[10] manages digitally acquired images, *Dentofacial Planner*[10] is a powerful cephalometric analyser, ...

It was always believed that going “paperless” (i.e. replacing all paper in the practice with computers) would allow the orthodontist to work more efficiently. Although this has proven to be true, a lack of organization and patience in developing such paperless solutions introduced a series of negative side-effects. In fact, many companies have tried to

provide a solution that would encompass/attend all necessities, but nobody was able to fully succeed: some clinicians prefer one solution more than the other, others wish they could use software A to just do what it performs well in, and use software B for doing the rest, others simply can't even dream of using the software they would like because it would require them to replace all of their existing hardware in their practice. It often happens that one practice ends up purchasing and using various software packages from different companies, requiring individual maintenance and, in some cases, keeping two copies of the same data! In the end, has the computer really simplified the work, or has it just added more stress, complications and practice downtime?

The problem arisedarose during the fast development of the computer industry, when the desire of using computers everywhere and for everything caused a great demand for software packages. This placed pressure on software companies which started developing software in order to quickly satisfy demand without paying attention to the big picture. Now that some time has passed, it is easier for us to take a closer look at what went wrong, and how it can be corrected: a standard is needed that unifies all software packages in one format.

1.3. The Solution. During our research, no software package was found that would provide a solution to the three basic issues listed in Section 1.1. Take a look at Table 2 to see a list of available software and how each satisfies the three problems. We believe these to be very serious issues which require immediate attention and envision PANIO to be able to provide the orthodontic community with the ultimate solution.

PANIO is not a program. It is a set of regulations that define how programs should be written and a set of shared libraries that aid program development. PANIO plans to attend the above mentioned problems by:

- (1) *providing a modular system.* All common tasks (i.e. scheduling and patient data management) are gathered into one program, while all more specific tasks are handled by third party modules. This differs from current modular programs, in that PANIO *depends* on third party companies, instead of providing modules (e.g. OrthoTrac [6], Visual Care System[8]).
- (2) *being platform independent.* Implementing PANIO and all of its modules in a software independent language, such as JAVA[5] or Visual C#[1], removes bonds with any particular hardware architecture. Systems like QuickMorph[9] or Advanced-Ortho[7] achieve this with an Internet-based system. Although we do intend for PANIO to have an Internet module, it is planned for it to install as a regular program on the local machine/network.
- (3) *providing a set of orthodontic-specific developers libraries.* Module writers will no longer need to rewrite code, because they will be able to take advantage of all of the core functionality of the system, which will be freely available under a public license such as the GNU Public License (GPL)[3]. They will only have to focus on writing the module.

2. THE DETAILS

Our intent is to help the orthodontic community, and not to impose a supposedly “better way of doing things”. We do not want to compete on the commercial market. We want to simplify the development and the use of orthodontic software. It is therefore of fundamental importance

for every user to have a choice of experimenting with PANIO: it's core program along with documentation and all of it's libraries should be freely available for download over the Internet. Anyone should be able to obtain and commercially use a copy that can handle all basic practice management tasks for free.

Similarly, program developers should be able to obtain a copy of the source code, all the libraries and all of the systems definition and documentation for free in order to encourage the writing of PANIO modules. These premises lead to making PANIO a free open-source project.

Keep in mind that these premises do not imply that all third party modules and core system implementations must be open-source and free as well: anybody is encouraged to write a module or even to rewrite the core system and sell it commercially at any price, as long as they adhere to the standards and the license set by PANIO.

The development of PANIO can be broken down in three main stages: **assessment, modelling and implementation.**

2.1. Assessment. In the first stage we will develop contacts and collect information from professionals in the field that have had extensive experience with using, developing and/or researching with orthodontic software. We want to make sure PANIO will be well accepted by the community, and, in order to be so, it is necessary to be in constant touch with other professionals.

During the first phase data will be collected from interviews with collaborators (clinicians, research orthodontist and software developers). These interviews will provide data that will allow us to:

- (1) define the boundaries of the system;

- (2) make sure the presented solution will be well accepted the orthodontic community;
- (3) revise the initial goals in order to better match the consumers needs;

The main project participant is Prof. Franco Magni who will select further participants.

Once a good number¹ of interviews has been completed, the information will be organized and used to form more specific project goals and boundaries at which point a first draft of the program definition can be worked on. Although the first phase will be technically completed, interviews will continue to be scheduled since the project depends on the constant contact with and feedback from other professionals. This is expected to happen throughout the whole development of the project.

2.2. Modelling. The modelling phase consists of developing the core of *PANIO*: its definition. This is the actual project as all consequent software development will follow the strict rules set in this stage.

To develop a model is not an easy task. For this reason we intend to extend already well developed efforts such as HL7[4] and OpenEMed[2]. More extensive research will be done to verify which is the most apt for the orthodontic software community.

2.3. Implementation. A program will be developed to implement the standards set by *PANIO* in order to prove and test it's functionality and to create a starting point for further developers.

Open source project research (references) suggest that it is difficult to form a project community using open source portals if the project is

¹This number depends on the quality and time of each interview: a few interviews with very rich content might allow the project to move on to the next phase at an earlier date.

posted too early. Possible contributors are not interested in ideas; they want to be able to see something actually working. We do not therefore expect the to gain popularity until a program has been released.

It becomes clear why simply writing PANIO as a set of definitions would not be enough for the project to gain popularity, and the importance of its implementation..

posting a project on an open source portal at a very early stage, does not To do this, a new model will be written, to specify the functionality of the program. The model will then be implemented. Programming help in this phase would be greatly appreciated. To obtain programming help, at this phase the project will be released on an open source portal such as sourceforge. We hope to be able to gather a community interested in the project by the time a first release of the program has been completed.

Once a first core implementation is released, the project will be opened on an open source project portal, such as SourceForge[ref]. (It is not realistic to expect developers to start contributing to an open source project at the very beginning.)

2.4. **Time line.**

- (1) Assessment (60 days);
 - (a) Collect information through interviews;
- (2) Write definition model (12 months);
 - (a) Define project boundaries;
 - (b) Investigate different modular architectures;
 - (c) Investigate which definitions to use (HL7, ...);
- (3) Implement definition (12 months);
 - (a) Write implementation model;

- (b) Investigate which software platforms to use (what JAVA version, what database, ...) ;
- (c) Distribution/Testing: open project to open source portal on Internet.

2.5. Required Resources. The entire development of *PANIO* does not require any major infrastructure or investment. The only required resources are:

Travel funding: Collecting information means interviewing with professionals all around the world. While many interviews can be done via telephone, email or video-conferencing, others might require physical presence.

Computer with Internet: access will be required in order to complete all subsequent phases of the projects. Since writing the definition of the model and the actual model is part of a masters thesis, the University of Brasilia will provide this resource for the first year;

Books and other references: will be required. When possible free resources will be used (as in public libraries and Internet), but purchasing a textbook might be the only solution to some problems;

Programming help: will be needed in the case a deadline is to be met. This means paying for programmers to work on the implementation phase;

3. CONCLUSIONS

We believe the orthodontic community can benefit greatly from a solution as *PANIO*. *PANIO* offers a standard for programmers to adhere, a base for all orthodontic software development. It is a meeting point for

all developers and users, such that the progress of orthodontic software can grow

We believe that by providing a standard and a base for all orthodontic software, new software can be developed and released at a faster rate, increasing the productivity of orthodontists. By doing so, PANIO's goal is to simplify the life of orthodontists, and of program developers.

Since project is relatively long term (see Section 2.4), we do not expect to have a usable product for the first two years of development. Nonetheless we do expect to form a community already in the first stage of development that will greatly help and determine the future of PANIO. It is this same community that PANIO depends on: since it is a non-profit effort, it's development is fueled by the enthusiasm of possible users.

APPENDIX A. AVAILABLE SOFTWARE TABLE

Name	modular	platform- independent	open standard
Dentofacial	no	no	no
ViewBox	no	no	no
Morph-E	no	no	no
Loop	no	no	no
OrthoTrac/-	yes	no	no
OrthoWare/OPMS			
QuickMorph/E-	no	yes	no
Morph/ODTP			
OrthoVisionTech	no	no	no
Pordios	no	no	no
dolphinimaging		no	no
Onyx-Ceph	no	no	no
Trace-X	no	no	no
Visual Care Suite	yes	no	no
Dr. Ceph/Dr. View	no	no	no
WinCeph	no	no	no
QuickCeph	no	no	no
Z1 KFO	yes	no	no
OrthoEase	no	yes	no
Healthware for Dentists		yes	no
Dental Office Manager	no	no	no
VistaDent		flash page	no
Program Director	don't think so	yes	no

TABLE 2. In this table we report how the researched softwares attend the three basic problems described in section 1.1.

REFERENCES

- [1] Microsoft Corporation. Visual c#. <http://msdn.microsoft.com/>, 2003.
- [2] David Forslund and Jim George. Openemed. <http://openemed.sourceforge.net>, 2000.
- [3] Free Software Foundation Inc. Gnu general public license. <http://www.gnu.org>, 1991.
- [4] Health Level Seven Inc. HL7 standards. <http://www.hl7.org>, 1997.
- [5] Sun Microsystems Inc. Java[tm] 2 platform. <http://java.sun.com>, 1995.
- [6] Kodak. Orthotracc, orthoware, opms. <http://www.kodak.com/>.
- [7] Charles A. Lewis. Internet-based practice management software. <http://www.advanced-ortho.net/>, 2002.
- [8] Orthocom. Visual care suite. <http://www.orthocom.com/>.
- [9] OrthoMorph. Quickmorph. <http://www.orthomorph.com/>.
- [10] Dentofacial Software. Planner, planner plus, showcase. <http://www.dentofacial.com/>, 2004.