1. THE ISSUE

HEALTHAPP is looking for a stopgap solution to improve the user interface, performance, and marketability of their flagship integrated hospital management application. This stopgap may eventually serve as the foundation for a complete rewrite of the program from scratch—after the company’s IPO. HEALTHAPP’s stated solution is to take a mockup front–end interface to the product—created in the Microsoft Visual Basic language—and further develop that code to serve as the interim interface to the product.

As marketing and management advisors, ARK Enterprises is concerned that a healthcare management application that relies in part upon Microsoft Visual Basic may be met with strong criticism and even boycott from industry critics and IT managers who purchase or recommend large systems. We are concerned that Visual Basic and products created from it suffer from several perceived problems among Information professionals.

This is not a technical issue of what the Visual Basic language can and can not do, it is a marketing consideration similar to the difficulty of marketing food which contains artificial sweeteners.

1.1. RECIPE FOR DISASTER?

As a consumer, I’ll buy anything that lists sugar or corn syrup without a second look at the ingredients. When I encounter a product listing saccharin, sorbitol, Nutrasweet, aspartame, or other no–calorie substitutes among the ingredients, the purchasing decision becomes more difficult. My mind races with half–baked recollections and warnings... “Didn’t someone die from eating sorbitol? Doesn’t saccharin cause cancer in rats? Will aspartame and Nutrasweet make me go crazy?” The FDA has formally evaluated and approved the consumption of these chemicals—they would probably not affect my life expectancy by one minute. Regardless of the truth, I am biased against purchasing and using these artificial sweeteners, and probably will be for the rest of my life.

What if HEALTHAPP’s software product contained an ingredient that made people aversive in a similar fashion?

2. THE USENET POST

In an effort to widen my understanding of the social perceptions surrounding programming, purchasing, using, and relying upon Microsoft Visual Basic–based products, I dropped an open–ended message into the Usenet. The Usenet is a loosely organized body of over 18,000 topical discussion groups. The message summarized HEALTHAPP’s general software project and asked four specific questions.
2.1. NEWSGROUPS SELECTED

Here is a list of the specific Usenet discussions where this message was directed:

1. comp.lang.basic.visual
2. comp.lang.basic.visual.misc
3. comp.lang.c
4. comp.lang.c++
5. comp.lang.cobol
6. comp.programming
7. comp.soft--sys.powerbuilder
8. comp.software--eng
9. comp.sys.hp.hpux
10. comp.sys.hp.mpe
11. comp_unix.programmer
12. microsoft.public.industry.health
13. microsoft.publicvb.enterprise

These groups were selected as a sample of programming and healthcare technology discussions where such a blatant request for free consulting would be tolerated. HEALTHAPP can probably extend the usefulness of this survey by having its own representatives post more specific queries to the listservs and newsgroups in which they regularly participate (and where, as regular, recognized, and respected participants, their direct queries will not be interpreted as a rude request for a free handout).

2.2. MESSAGE, AS POSTED

Designed to be slightly vague (prevents responses hung up on technicalities) and slightly sensational (provoke more responses from a world wide, yet sometimes apathetic readership). Details for retrieving the public responses in this discussion thread appear at the end of this document.

Greetings Netizens with Program/System Design expertise:

Has Visual Basic matured to the point that it can be used in a mission-critical healthcare application?

A major vendor of healthcare software is trying to decide if they should revise their flagship application using Microsoft Visual Basic.
This 'mission-critical' application is presently implemented in Micro Focus COBOL on HP-UX/MPE servers, feeding dumb-terminals throughout the hospitals in which it is installed.

I'm under the (mistaken?) impression that Visual Basic has a stigma of being 'inappropriate' for developing mission-critical, realtime systems filled with sensitive patient and financial data. If so, it may be awfully difficult for us to pitch IT-savvy Hospital IS managers and CEO's on the product (we provide this client with marketing and management services).

Your experience, advice, answers and anecdotes regarding these four points will be read with great interest...

1. What's the 'social perception' of VB among IT professionals?
2. What's the technical viability of using VB for a healthcare app?
3. Any alternative solutions you suggest we look into?
4. Would you trust your life to a hospital that runs on Visual Basic?

TIA, anyone who has a minute to help us out by sharing their $.02

--Sean Dreilinger
please communicate any blatant ads/advocacy/flames via private e-mail

Sean Dreilinger, MLIS
PGP Public Key - http://www.kensho.com/sean/pubring.htm
sean@kensho.com - 619.514.3939 - http://www.kensho.com/~sean/
KENSHO - Bringing Knowledge to the Information Age - in a Flash

3. SUMMARY OF RESPONSES

I’ve put together the results in two ways: a graphical representation and a collection of interesting quotes.

3.1. GRAPHICAL SUMMARY

From each response, seven questions were subjectively answered and graphed:
Implications Of Visual Basic For Healthapp

General Tone of Response

Does Visual Basic have a negative connotation among IT Pros?

Is VB Technically Viable for This Application?

Identified Specific Limitations of VB

Mentioned Personal Experience With VB

Offered Alternative Solutions?

Do You Trust VB In A Hospital App?
### 3.2. QUOTES OF INTEREST

If you're not going to wade into the 58 in–depth responses, the following excerpts may be of interest.

<table>
<thead>
<tr>
<th>NAME</th>
<th>ORGANIZATION</th>
<th>THOUGHTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;KJS Systems Group Ltd.&quot; <a href="mailto:kjs@oxford.net">kjs@oxford.net</a></td>
<td></td>
<td>My Company is in the process of moving our very successful DOS and UNIX based Financial Management /Inventory System from prior versions of Microfocus to a Visual Object Cobol / Visual Basic combination. Working really well.</td>
</tr>
<tr>
<td>&quot;PLeckett&quot; <a href="mailto:leckett@magi.com">leckett@magi.com</a> PWGSC / TPSGC</td>
<td></td>
<td>Only 16% of client/server apps are completed on time and with the functionality required. I am sure this dismal figure is due to a large part in not picking the right tools to fit the job. The issue of whether to use VB or another product such as Powerbuilder, Delphi or C++ is really dependent on a number of issues and cannot be decided just &quot;because someone likes it and everybody else is using it&quot;. The point I am trying to make is the selection of a product is far too complicated to be resolved over a newsgroup. Take my advice. Get the best impartial application architect money can buy. In the end you'll more than re-coup your costs as compared to get bogged down in a never ending development project.</td>
</tr>
<tr>
<td>&quot;Paterson, Robin&quot; <a href="mailto:robinp@buntypost.dundee.ncr.com">robinp@buntypost.dundee.ncr.com</a></td>
<td></td>
<td>However, what you absolutely must realise, is that because several files are generated for each app (source, forms, etc), this will require <em>major</em> maintenance. Yes, using the GUI you can throw up <em>prototypes</em> very quickly, but any <em>proper</em> code development will take much longer. You must ensure management understands this !!!!!!!!! Just because the front end looks not–too–bad, doesn’t mean the job’s done.</td>
</tr>
<tr>
<td>&quot;Peer Tvrngren&quot; <a href="mailto:petn@im.se">petn@im.se</a> IMI (Industri–Matematik International)</td>
<td></td>
<td>I would feel very unsafe about it if I saw the VB logo on the PC by my bedside. All in all, I have a strong feeling that VB systems in general are implemented by optimistic amateurs with a naive “this can’t be so hard” approach. It is by no means always the case, but I think you will get that type of immediate reaction from a lot of people.</td>
</tr>
<tr>
<td>Adam Nikic <a href="mailto:adamnik@daemon.apana.org.au">adamnik@daemon.apana.org.au</a></td>
<td></td>
<td>Having run that test application I have found that the application developed in DOS (QuickBasic 4.5) is even faster on old IBM 286 than the same on 486 under Windows environment. So my conclusion is obvious to go and do all stuff in DOS! I do not know whether you heard of disassembler for VB3 for WIN. I tried it and really it made source codes out of EXE file. Something like putting sausage in that disassembler and get cow at the end of the process. So people who knows that can mess with my code and I cannot recognize it. Maybe in critical moment of testing I will recognize that something is going in wrong way. Huh, it would make me nervous and suspicious on VB3. Greetings from engineer who is fighting for machine 'life'. I wish you best in fighting for human life. Remember, I can always make excuse if my machine dies, what about human?</td>
</tr>
<tr>
<td><a href="mailto:alfiesty@rmii.com">alfiesty@rmii.com</a> (The Alfa Kid)</td>
<td></td>
<td>His is sort of loke writing apple.com and asking about the utility of MSWindows. do you REALLY expect a real non-biased reply?</td>
</tr>
<tr>
<td><a href="mailto:algie@tcp.co.uk">algie@tcp.co.uk</a> (Alan Davis)</td>
<td></td>
<td>I am, almost running VB down, I am a Windows Designer/developer, whose percentage of developing time is 5% C++, 15% Delphi, and 80% Visual Basic!</td>
</tr>
<tr>
<td><a href="mailto:buster13@indy.net">buster13@indy.net</a> (Ken)</td>
<td></td>
<td>Look in any issue of Visual Basic Programmers Journal and you’ll see some pretty amazing applications under the section called &quot;Basic Heroes&quot;</td>
</tr>
<tr>
<td><a href="mailto:davidb@datalytics.com">davidb@datalytics.com</a> (David Bradley) Datalytics Inc.</td>
<td></td>
<td>All too often the reason to use BASIC is so you can find cheap people to write your software. And that's what would make me stop and think. Did they also cut corners in other areas. Maybe if they choose to cut costs on developers they also chose to cut costs on testing and quality assurance. I've been involved in a couple of VB projects that turned into monsters because the complexity was too great for VB. The time I saved up front was spent on the back side trying to wrestle VB to do what I needed. Usually something that would have been trivial in another language.</td>
</tr>
</tbody>
</table>
COBOL would be on my list well before VB, or even C or C++. If your people feel comfortable working in COBOL, that may be the best solution. If I found out an x-ray machine was programmed in VB, I would not be in the same room as it.

Bad design of the database (or other components) architecture will destroy performance, and VB often (and unreasonably) gets the blame for this. VB is too easy to use for safety. It encourages some extremely poor software engineering practices amongst VB "experts" who simply don't have the depth of knowledge of complex systems to be capable of putting a big system together, no matter what it's written in.

And make sure you have real professional programmers, who understand things like structured design and VB error handling. Too many VB programmers just drop some controls on a screen and think they're done.

There are more solutions than this, as long as you are willing to use more than one tool. But if you want to crank out a "soup-to-nuts" mission critical app using only VB, you probably shouldn't waste your time.

You should take a look at the article "When opposites attract" in the June 1996 issue of Object Magazine. It outlines a strategy for using Visual Basic as the rapid prototyping/GUI portion of an application while using Visual Object COBOL for the business model. This makes particular sense in your situation because you already have your business model in MicroFocus COBOL. The Basic and COBOL can communicate through DLLs or OLE. It also makes sense from the viewpoint that you can utilize your existing staff for the conversion because they are presumably skilled in COBOL and just building GUIs in VB is easy.

there ain't no chance in Hell that I'd want to depend on it for something critical... I've worked with it too much to trust it for something like that.

Let me say one thing. Although VB is a fine language to use in small apps, it will be a cold day in hell before I trust my life to a hospital run on such an app.

The thing that will kill the safety of your client's product, more than anything else, is a traditional approach to software development. You can't just code-and-test; you need to design and plan first, then do code.

I am very under impressed with VB, I've recently had to debug some VB3.0 code. It is difficult to navigate projects with it in my opinion... I would never trust my life to any current Microsoft product. They are often well designed but flaky underneath. As long as you want to write your app exactly as MS would like you to you'll be fine. Straying from the MS route will result in many tears due to a lack of flexibility.

My dad is a doctor here in Sweden. I have been looking at the system they are using and it's just terrible. It is programmed in a VB like language... I could just see the "system crash" message in the emergency room...

Do not underestimate the amount of effort it will take to develop a robust VB application.

I have managed a number of projects using different languages. I feel that its NOT the language, its the programmers. Unfortunately, most of the BASIC programmers haven't a clue how to modularize code due to lack of WANT or lack of correct
programming practices. I have found with many programmers it is the lack of WANT and TIME. BASIC programmers generally come from a background of 'just get it done.' If I find a BASIC programmer that doesn't have that mentality, I grab them and then help them into other Languages. Modularization is the KEY to mission critical applications (my opinion). Modularization is also the key to cost effective upgrades and the never ending change of requirements.

Peter Seebach  
<seeb@taniemarie.solon.com>

I would never, ever, *ever* risk my life on a Microsoft language product. The last one they did that wasn't horribly buggy was BASIC for the TRS–80.

Philip Staite <pstaite@VNET.IBM.COM>  
IBM, Rochester, MN

I still think of it as a toy language, good for GUIs, but not much else.

Randy Neff  
<rneff@pequod.Teknowledge.COM>

Microsoft is famous for changing the APIs of their interfaces every year. So the biggest risk of using Visual Basic 4.0 is that everything will have to be rewritten for VB 5.0 and VB 6.0, since VB 4.0 will not run on Windows 97 or 98 or whatever. Microsoft, both at the OS level and at the language level, is a continuously changing target.

Timothy Hayes <timhayes@novia.net>  
Novia Internetworking  
<2.8kbps dialup; 602/590–2NET>

data is data is data. It doesn't matter much what the data is

Vincentd@cyberspc.mb.ca (Vincent DiBernardo)  
Structure Software

I've used Hp's some time ago and have had little problems with MPE. However the support costs to Hp were a constant thorn in my side along with the notion of closed systems. I'm not quite sure what your company is trying to accomplish: VB front end to an Allbase database or a complete switch to NT server/ MS SQL 6.x and VB as the front end. A lot of companies are moving to client server because of perceived pressure to do so. Recent studies have shown that the investment has not paid off for a lot of companies and has yielded less throughput.

Walt Howard <howard@ee.utah.edu>  
Electrical Engineering Dept, Univ. of Utah

Sensitive patient and financial data usually want to live on a system which has some concept of security. Most versions of Microsoft Windows don't even have the concept. Those that do, implement it badly. Unless you are planning to store this data with your own additional security measures (good encryption, for instance), your data will be vulnerable. (Donning flameproof suit) It is surprisingly difficult to beat a system that is well-coded in modern COBOL (or one of the 4GLs that translate to COBOL) using a solid and secure underlying database (such as the Image DBS on MPE). I think that in order to provide a PC-based system which is truly equivalent to the one you want to replace, you will have a hard time beating the price and a really hard time beating the tasking performance, and an impossible task to beat both. When I did an equivalent but much smaller project here, I used C instead of COBOL (I know it better, and I did not need the advanced features of COBOL), and it ran rings around the Microsoft–based solution, even though the hardware was slower. The user interface did not look as nice as what you will do in an afternoon, but it was clear to the user and required minimal input from the user to do useful work. And it was very maintainable. But it's hard for me to believe that VB could be fast enough for a system with lots of data, and I absolutely know that hacking such a system would be too easy for too many people. So I would have no confidence that my confidential patient records would really be confidential. And I would worry that, if the computer kept my patient charts, they might go incorrect at any time. Keep in mind: drug abusers like to (try to)
get jobs in hospitals, because it is easier to steal drugs there than from a hardware store. So there will be potentially hostile insiders with access to the computer system. You need assurance that they can do limited damage, and VB doesn't provide much.

Zach Heilig  
<zach@freebsd.gaffaneys.com>

You should also think about security. Unless you run the VB app under WinNT, there is no security at all. Anyone familiar with the Windows environment can easily bypass anything you can throw in their way with VB. From my experience with programs written directly by Microsoft, I definitely wouldn't want my life depending on their reliability.

3.3. THE STIGMA

Responses confirm that while Visual Basic (along with top-notch programmers and careful planning) is technically capable of doing the interim programming job (at least as far as they could understand from my over-simplified sentences describing it), The application and its use still suffer from a negative perception among IT professionals. Those "VB could–do it" answers are often tempered by mention of technical glitches or limitations of Visual Basic that didn’t apply to the respondents own work but that might be relevant (and devastating) to HEALTHAPP. At the same time respondents were agreeing that VB is technically capable of doing the job, they also reinforced the negative stereotypes associated with the program, its use, and its users:

- Microsoft makes slow, unreliable software and changes ‘standards’ often, forcing software programmers to re-write code in order upgrade and use new features.
- The basic language from which VB evolved is a beginners language
- VB is interpreted, and all interpreted languages are too slow for ‘real’ projects.
- VB is insecure and easy to reverse engineer
- VB attracts dumb programmers

For the purpose of marketing, the validity of these stereotypes is less important than the simple fact that they exist and potential customers may be swayed by them. Having these clouds of “Visual Basic doubt” lurking around HEALTHAPP’s interface will make it hard to pitch to customers, who may easily subscribe to one or more of these biases, regardless of the product’s technical stability and credibility.

4. SUGGESTIONS

Give the contributions of these Internet users from around the world some serious thought. They have no personal stake in the development of your product, and many of the pro–Visual Basic respondents were extremely cautious in recommending use of the tool for large–scale mission–critical installations. Several responses cited articles, books, products and services that may provide a helpful alternative to Visual Basic.

HEALTHAPP is not an internal corporate application where management can dictate what employees must learn to love and use.

HEALTHAPP is a major commercial application with a very high price tag. Customers shop for it via RFP’s and scrutinize their purchase options very carefully. HEALTHAPP system users entrust it with the life-information of their own clients—patients. If the hospital IT staff and purchasing decision makers suffer from any of the potential biases against Visual Basic, HEALTHAPP sales will also suffer.

The purpose of this adventure was to provide a wide–perspective opinion of HEALTHAPP’s present programming direction and the way it may ultimately be perceived in the community of Information Technology professionals who will be involved recommending or denying purchase of a HEALTHAPP installation for their own hospital.
5. ADDITIONAL INFORMATION

An up–to–the–minute array of publicly–posted responses from all 13 Usenet discussions should be available through mid–August by visiting DEC’s Alta Vista Search index via the following prepared query–URL, which contains no line breaks or blank spaces:

http://www.altavista.digital.com/cgi-bin/query?pg=q&what=news&fmt=.&q=%22Trust+Your+Life+to+this+VB+App

About 58 responses have been received via private e–mail and public news to date, with more trickling in. The discussion threads had been going for about 72 hours at the time this document was prepared.