Does Computer Science Need Computing?

The Science Behind IBM Wimbledon

Matt Roberts

IBM Hursley Laboratories

Anyone for Tennis?

The Science behind IBM Wimbledon

Overview

This document provides a verbose text version of the PowerPoint presentation of the same name, which should hopefully more readable than a set of slides!

Contents

Introduction

This presentation provides some examples of how science and IT interact with each other (and how IT is used in sport) using IBM Wimbledon to provide some examples. Hopefully it will help you to start thinking about the types of problems that are found in IT and how they can be solved.

IBM Wimbledon

The first Wimbledon Tennis Championship was in 1877, and IBM has been the "Official Supplier of Information Technology and Consultancy" since 1990. IBM's involvement at Wimbledon is all year around, but centred on the two weeks of the Championship itself.

IBM are responsible for not only the Wimbledon website and online shop, but a complete information capture and delivery system used by spectators, umpires, press, TV stations and commentators, radio stations and more. The system handles every match point, statistic and news item that takes place during the tournament.

The event is about as high profile as you can get. Around 470,000 spectators visit the Club itself to see the tennis first hand and there are around 2 billion TV viewers in 159 countries. As you can imagine, this is not the type of event where you can tolerate something going wrong. Imagine the bad publicity that would be created if the computer systems failed during the fortnight, or weren't ready in time for the start of proceedings – you can't delay the start date because the computers don't work!

The All England Lawn Tennis Associated (AELTC) have chosen IBM for the last 14 years to be their IT partner because they know that the company has the skills and experience to put together the necessary hardware and software systems required to run the event smoothly.

IBM manages a team of around 150 staff for the Championships. The team are responsible for the TV graphics, radar speed of serve machines, on-site information display content and boards, data capture of the individual statistics from each match, a large network of computers across the site, entering and editing the website content as well as the basic project management activities.

What's IT all about?

Information Technology is all about problem solving. When you look at it, this isn't just limited to IT, but in fact any kind of business. The best paid jobs are where people are paid to use their intelligence to solve business problems.

Almost no company uses IT for the sake of it (some do, but they are the exceptions to the rule!). In the vast majority of cases business employ IT in order to gain some kind of competitive advantage – usually by improving efficiency of their existing processes, or by enhancing their products or services.

Jobs in the IT industry

There are many different types of jobs in the IT industry. The other presentations in this lecture series are focused on developing new products of some description, whether they are hardware or software, and around 2,000 people at the IBM Hursley site work in this kind of role. However there

are another 19,000 IBM employees in the UK who don't do this kind of job, and they work for IBM Global Services (IGS) – the consultancy arm of IBM.
Consultancy (whether it be for IBM, Accenture or any of the other consultancy firms) essentially consists of the customer employing you and your colleagues to carry out a piece of work for which you have the necessary skills, but they do not. For IBM this is usually creation of new IT systems and integration of existing systems on behalf of their clients. Often this involves integration of IBM hardware and software products, but it could equally not involve any IBM products.

Jobs in consultancy (for example with IGS) tend to be customer facing, with employees required to dress smartly in suit and tie, taking a high level overview of the way in which different systems integrate with each other. By contrast, development style jobs like in the Hursley Laboratory are usually working on products that will hit the market in 12 to 18 months time, and don't face directly to customers. This means that development employees usually dress more casually (jeans and t-shirt is common at Hursley) and work to different deadlines. Consultancy employees are the frontline breadwinners for a company – it is their work that brings in the cash, but without the development employees they would have no new products to offer to customers.

What skills do you need?

Graduate recruiters for IT firms look for a variety of skills in their potential recruits, including the following kind of things;

- How you interact with other people – teamwork is a massive part of the business environment.
- What are you like when you're working under pressure? (to deadlines, with new colleagues etc)
- How good are you at solving problems?

What this list doesn't include is knowledge about IT systems. While for some jobs specialist knowledge is useful, most businesses would much rather take a new employee with the right foundation skills (as described above) and teach them the job specific (IT) knowledge and skills.

What percentage of IBM graduate recruits in the last 4 years do you think have come from Computer Science degrees?

You might be surprised to know that the actual answer is only 1 in 3. **To rephrase that, 2 out of every 3 new IBM employees don't have an IT degree**. In fact 1 in 6 new recruits have Science degrees (Physics, Chemistry, Biology) with a further 1 in 6 from Business degrees, and 1 in 12 from Maths and Engineering respectively.

Joining the physical and non-physical worlds

An important part of any IT system is where the information comes from – how does data get into your system, and how (if at all) does it get out again?

In many cases the source of information for your system is the physical world. This might be using a mouse or keyboard, but can also be from a variety of other input devices;

- Touch screen
- Traffic monitoring sensors
- GPS locating for displaying position on maps or locating trains on the rail network
- Barcodes or radio frequency labels to allow stock tracking

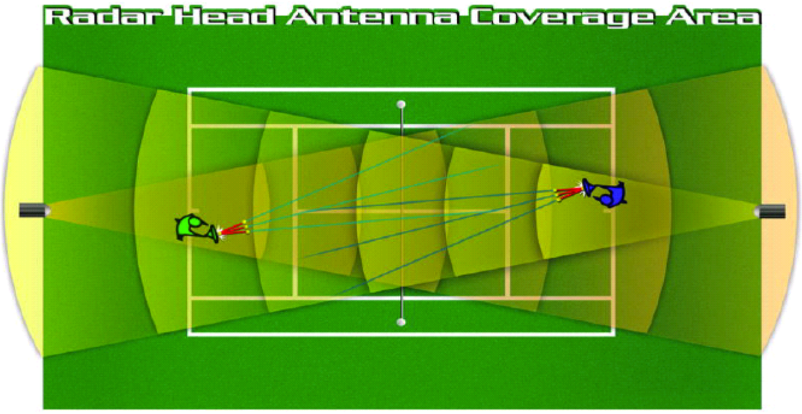
Equally there are numerous ways to get information out of an IT system, each of which is useful for a different purpose;

- Computer screen, printer, email, text message
- Braille screens (for the visually impaired)
- Audio output like automated announcements (for example at train stations)

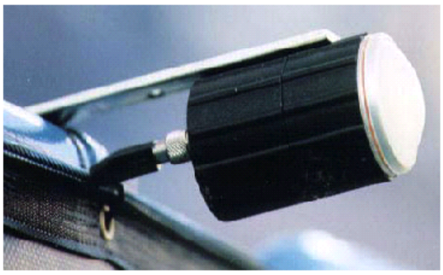
In the future, wearable computers or implants will be able to provide feedback from computer systems directly into the human body (for more info on this read “I, Cyborg” by Kevin Warwick – a lecturer at Reading University).

Speed of serve machine

One very visible part of the IBM Wimbledon project is the speed of serve machine. This uses a radar sensor like that used for measuring the speed of cars on the road to measure the speed of the flying tennis ball when it is served.

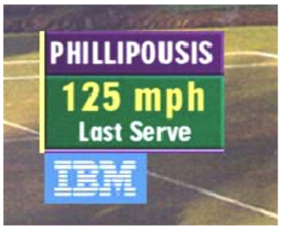


Radar head located at each end of the court



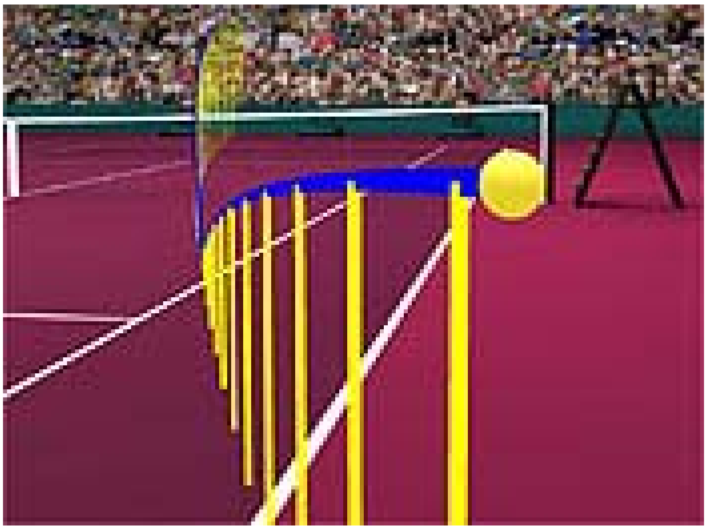
On court displays

TV Graphics



Hawk-Eye

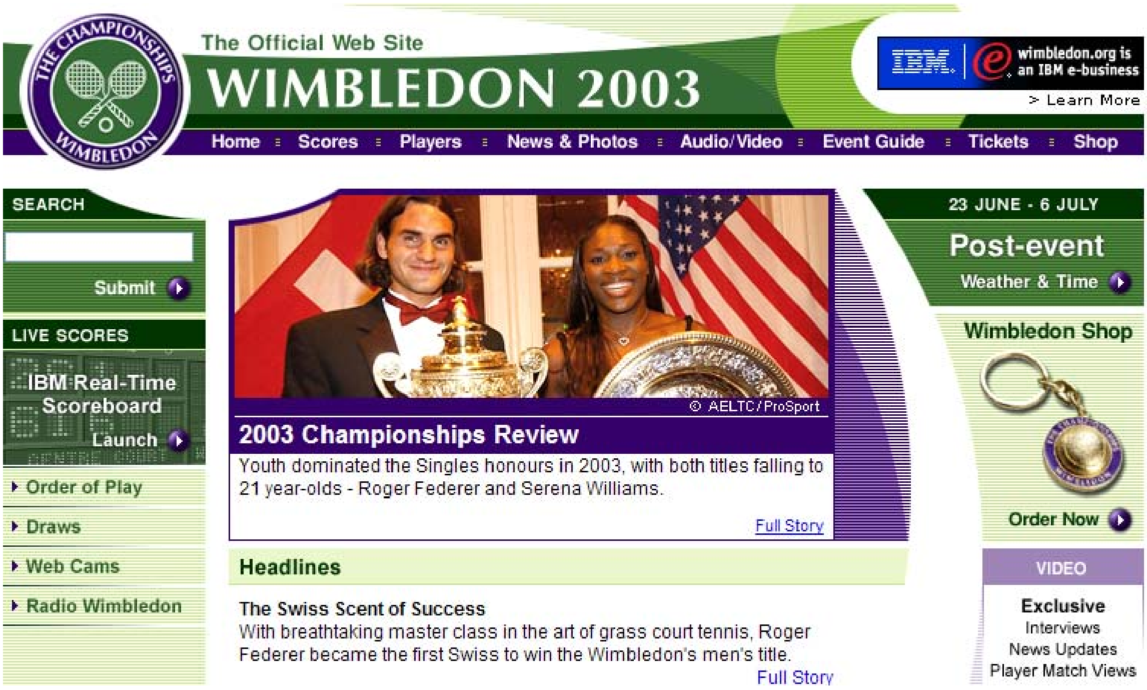
Another example of inserting information into a computer system is the application of complex computer software to model the flight of the ball in three dimensions. You may be familiar with the Hawk-Eye system used by Channel 4 in their coverage of test cricket. This year at Wimbledon the same technology was employed for tennis matches.



The system uses five cameras at in different locations around the court to observe the path of the ball as it flies through the air. The information from these cameras can then be combined using computer software to make a 3D model of the flight of the ball, which is accurate to within 2 to 3 millimetres. The software can create the model within 2 seconds of the end of the shot – easily fast enough for most uses.

TV commentators can use this to replay shots and second guess the umpire and line judges over whether a given ball was in order out, although there are plans afoot to allow it to be used as a tool for the umpire to get backup on controversial calls, a little like the TV replay option that is now available to referees in Rugby and Cricket.

Psychology at Wimbledon



The Official Web Site

WIMBLEDON 2003

wimbledon.org is an IBM e-business

> Learn More

Home : Scores : Players : News & Photos : Audio/Video : Event Guide : Tickets : Shop

SEARCH

Submit

LIVE SCORES

IBM Real-Time Scoreboard

Launch

- Order of Play
- Draws
- Web Cams
- Radio Wimbledon

© AELTC/ProSport

2003 Championships Review

Youth dominated the Singles honours in 2003, with both titles falling to 21 year-olds - Roger Federer and Serena Williams.

Full Story

Headlines

The Swiss Scent of Success

With breathtaking master class in the art of grass court tennis, Roger Federer became the first Swiss to win the Wimbledon's men's title.

Full Story

23 JUNE - 6 JULY

Post-event

Weather & Time

Wimbledon Shop

Order Now

VIDEO

Exclusive

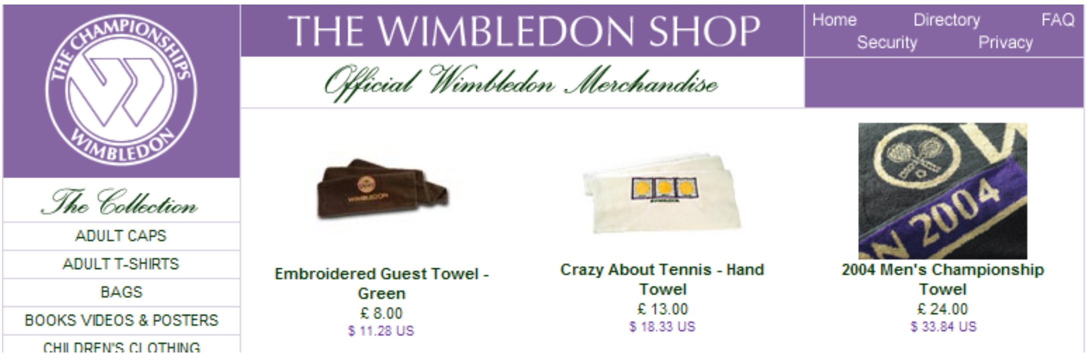
Interviews

News Updates

Player Match Views

The Wimbledon website provides access to all the latest match events, player biographies, photos, videos and statistics to viewers from around the world. It allows people from around the world to keep up with their favourite players and match ties regardless of which time zone they are in, and what they are doing at the time.

Some websites exist to make money (Amazon, HMV, LastMinute etc). This website is primarily intended to provide information to interested people – it isn't expected to pay for itself, you don't have to pay to look at the information. The aim of the website is to increase the popularity of the Championships by giving users access to as much information as they want to look at.



THE WIMBLEDON SHOP

Official Wimbledon Merchandise

Home Directory FAQ

Security Privacy

The Collection

ADULT CAPS

ADULT T-SHIRTS

BAGS

BOOKS VIDEOS & POSTERS

CHILDREN'S CLOTHING

Embroidered Guest Towel - Green	Crazy About Tennis - Hand Towel	2004 Men's Championship Towel
£ 8.00 $ 11.28 US	£ 13.00 $ 18.33 US	£ 24.00 $ 33.84 US

In addition to the information content of the site there is also an online shop where browsers can buy Wimbledon merchandise – branded caps, t-shirts and even tennis balls.

Because of the global appeal of the Wimbledon Championships the website is in use 24 hours day for the two weeks of the tournament – not just between 9am and 5pm GMT – but there are some time periods that are more busy than others (more on that later).

Ah, Mr. Jones – we have been expecting you

Consider the kind of issues that relate to any website – no matter how small or large;

Imagine you ran a bookshop in the Highfield area of Southampton:

- How many customers do expect to use your website?
- When do you expect them to use it? 24 hours a day, 9-5 only, 6pm-10pm only?
- When are the peaks going to be? During the day, evenings, weekends?
- How many computers do you need to cope with your customers? (probably only one!)

For a small bookshop you are likely to have customers from your own timezone only – people will be browsing at work and in the evenings, but there is likely to be very little activity overnight when everyone is asleep. The total usage will be low – you might get a couple of orders per day.

Now imagine you are the Head of IT for the All England Lawn Tennis Association:

- How many people do you expect to visit your site?
- When do you expect them to use it? During the day? Who's day? How long?
- When is the highest peak of demand over the fortnight
- How many computers do you need to cope with the demand?

In actual fact, there are 4.3 individual visitors to the Wimbledon website over the two weeks, notching up a total of 242 million webpages viewed. Each user stays for an average of 2 hours and 9 minutes, and they could come from any of 165 countries, so the demand isn't limited to 9-5 GMT.

As for hwen the highest peak is – take a guess?

- Mens final? (Saturday afternoon of the second week)
- Women's final? (Sunday afternoon of the second week)

The actual highest peak is Friday afternoon of the second week. It's the Men's semi final when both the UK and US populations are at work (and have free access to the Internet!). The peak demand for the Wimbledon website is around 250 times normal usage – how would you cope if you were given 700 questions to answer in a 3 hour exam rather than 3 or 4?

If you only have one server, then peaks in demand can overload the computer causing it to crash – then your website is not available, and your customers will not be unable to place orders or look at your site – making unhappy users...

One way of solving the problem is to make sure you have enough computers to cope with the highest peak in your demand. So if the Wimbledon website normally needs 4 computers to cope with average demand, they would need to buy 1000 computers in order to cope with the highest demand – expensive, especially considering 99% of them will be unused for most of the time.

One of IBM's solutions to this issue is to provide contracts that allow customers to pay for computers on the basis of how much they are used rather than how many you have. (This is one of the facets of what IBM calls "on demand")

Reponse times

How long will people wait for a webpage to load before they get bored?
How long would you wait? Would you wait for 10 seconds for every page to load, or would you get impatient? Would you wait longer for some items?

In actual fact the accepted maximum acceptable time for general webpages to load is about 5 seconds – if you can't get all or part of your webpage to display on your customer's screen within 5 seconds then they will get bored. Having said that, you might wait 10 seconds if you were waiting for an order confirmation, or a particularly important piece of information.

What happens if users get bored of waiting for your site to load? If you were trying to order a CD on the HMV website and you got bored of waiting for the page? Without any difficulty at all you can leave the HMV website and go to order the CD from Virgin. It's very difficult to engender brand loyalty over the internet, but it's really easy to lose it. If your website is difficult to use or doesn't respond quickly then you will lose customers to your competitors.

What could you do to improve the response time for users?

- More servers doing the same thing – spread the load across these machines so that each one only handles part of the load
- More servers doing different things – might have dedicated machines for gold card customers, or for people who are actually ordering things rather than just browsing. The people ordering items are more valuable to your site – you don't want them to give up ordering things because the site takes too long to process their order.

This issue and the previous one on handling demands on your website are basically about predicting peoples' behaviour patterns and knowing their tolerances – definitely in the realm of psychology at times!

IT Architecture the easy way

As I mentioned at the top of this document, business is all about problem solving.
What's the first rule of business?

The customer is always right

This usually means that the customer always gets what they want – but rule 2 (or 1b at least) is that the customer doesn't usually know what they really want even if they tell you what they think they want. They are however, very good at telling you what they don't want if you suggest something to them.

Because of this, designing IT systems usually consists of an iterative design process where you give the user a solution to the problem they gave you, and they decide it isn't what they want, and that they had the original problem specification wrong!

The aim of this section is to demonstrate that you don't need anything more than a passing familiarity with using computers to be good at designing an IT system. The problem we're trying to solve is as follows;

> *We want to provide Wimbledon fans around the world the ability to get scores from the tennis matches updated as soon as it happens on a point by point basis to encourage them to feel part of the action even if they aren't courtside...*

The first step is to gather requirements from the user. The following questions are a good set to start with;

- **Who are the target audience?**
- **How many of them are there?**
- **What information do they want?**
- **What does "as soon as it happens" mean?**

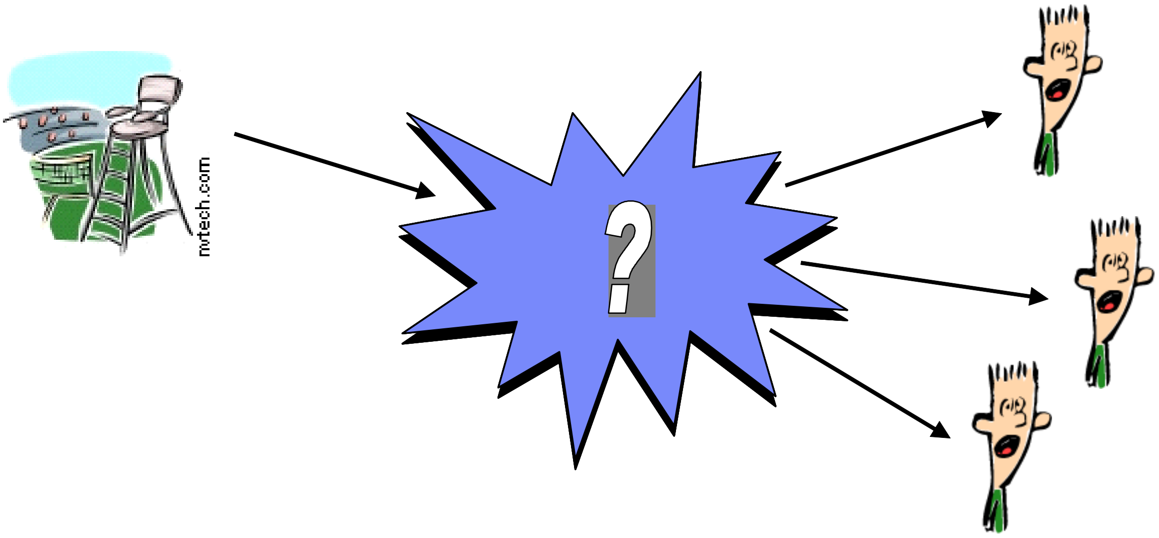
Obviously in this case the target audience extends out from the Wimbledon club to encompass anyone in the world, and the number of people we are trying to reach is probably some percentage of the 2 billion TV viewers (assume for simplicity that anyone who is interested in the championships will watch at least some of it on the TV!)

What information do they want? Updates for match scores. This is what the customer said in the problem specification – what do they really mean? Do they also want all the other stats information – speed of serve, number of aces etc? How much information does this mean we will be transmitting per match? There are 20 match courts in use during the championships – imagine a point is score every 20 seconds in each match – that's about a score update every second, without the extra stats information.

What does "as soon as it happens" mean? Do users want to be told about a change in score within a second of it happening? Would they mind if they only find out every 5 seconds? Every 30 seconds? By determining what the acceptable boundaries are you can begin to get a much better idea about how the project might work. Part of the answer will be based on how much the customer is willing to pay – the closer to "real time" you want, the more it will cost.

The Big Picture (System Overview)

Now we are going to develop a high level overview of the IT architecture for a system to fulfil the brief we have been given. It starts off looking like this;

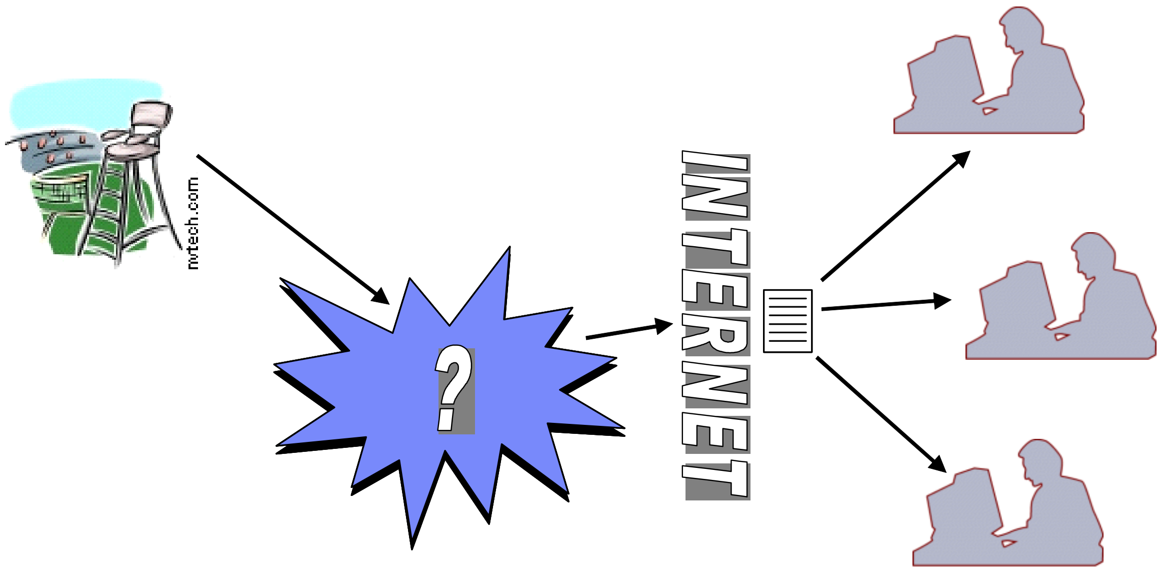


You know that there is some tennis going on in SW11 (the location of the Wimbledon Club), and you know that there are some people out there, but there is a big grey area (or blue for those of you watching in colour) about how it all happens.

How are users going to obtain the information?
Note that at this point it doesn't have to involve computers – the only reason to user IT is if it solves the problem.

- Post? Too slow, the information would be out of date.
- Phone? Too expensive – would need a lot of phone lines, user experience wouldn't be very good – they would have to sit with the phone glued to their ears.
- TV? Not a bad solution – note that it doesn't include masses of computer hardware/software. However for regular TV you can only see one game at a time (which someone else chooses) and you have to be sat in front of a TV set.
- Web Browser? Lots of people have access to computers, either at home or at work. The infrastructure for delivery data already exists. It doesn't cost you money to send out the information (like post would), only for the computers and network at your end.

Now the picture looks like this;



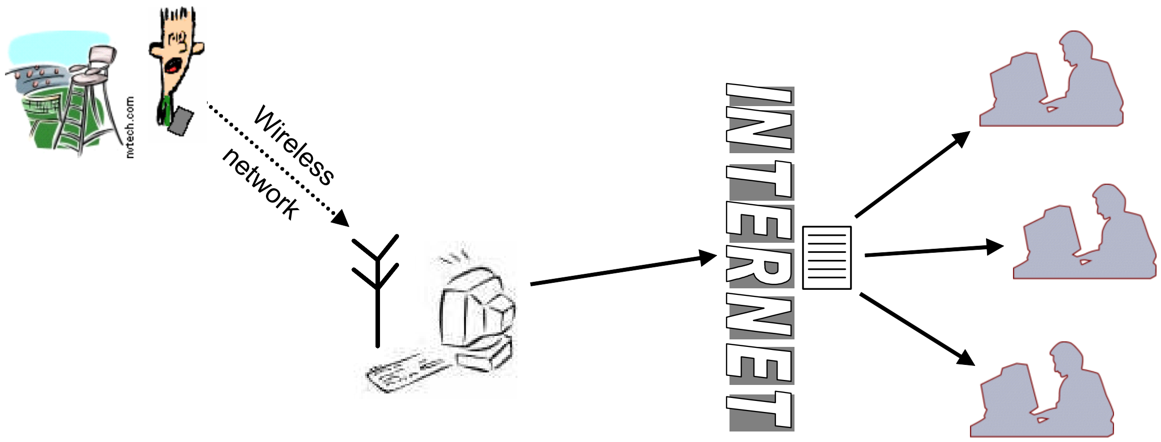
We’ve made the decision that users will obtain the information via the Internet, and you can see in the updated diagram that are users (now sitting at their computer) are looking at a webpage.

How does the information get onto the computers? We talked about this a bit earlier when we discussed the speed of serve machine and Hawk-Eye.

- There are 20 courts spread over 42 acres at Wimbledon.
- In order to provide a “real time” service you have to get the information from the courts onto a computer as quickly as possible so that it can be sent out to the tennis fans.
- Could build a big computer network across the entire of the Club and have a desktop computer next to each court wired in to the network. Would be very expensive and cause a lot of disruption for laying the cabling.
- How about a wireless network. Can use radio waves to transmit signals from the courts to a central computer server.
- Don’t want to have a bulky computer next to each court (at the very least would need a power supply). How about using PDA devices (PalmPilot etc) with wireless networking?

In the real IBM system there are a team of 40 county level tennis players who record the statistics for all the matches using PDAs linked using a wireless network. Information recorded includes the score, number of aces, double faults, how the points were won etc (ie all the information that you see commentators bringing up).

Taking this into account we get the following picture;



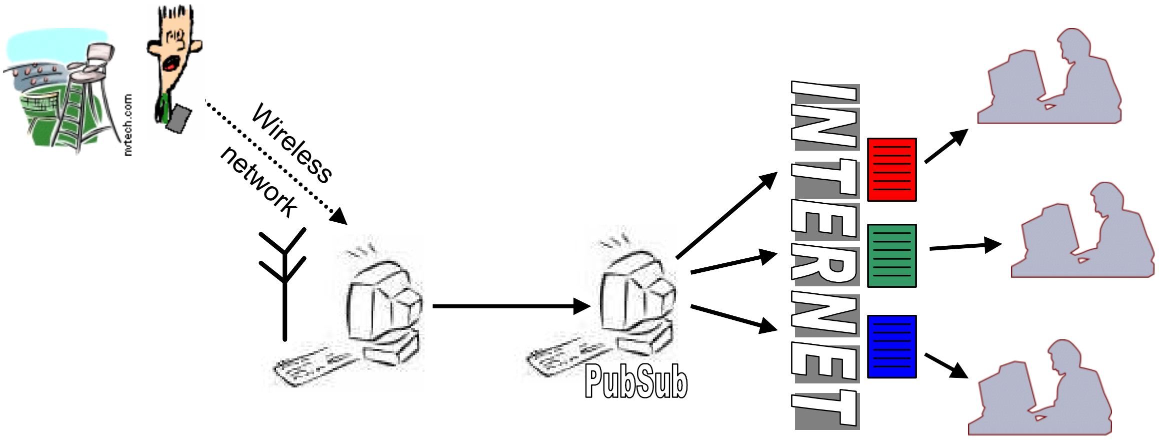
Now you can see our people at the Club with their handheld devices transmitting information to a computer using a wireless network onto a computer, which provides a webpage for users to look at.

How can we give the service a personal touch?

- A fixed webpage is very easy to do – someone could manually update the page every 5 minutes with the current scores, then people could reload their browsers to see the updated scores. Doesn't look very interesting, and the information is old by the time you see it... Advantage is that you only have to provide one webpage for everyone, but it doesn't give a very personal experience.
- It would be much better to give each user an individual experience, which means that you have to communicate with them all individually. This can be achieved using a technology called Publish and Subscribe, where people register an interest in the topics that are of interest to them. This way they can pick the match(es) they want to watch and leave out the ones they don't care about.

Another advantage of PubSub is that you can push the information onto the computers of your audience rather than them have to refresh the webpage to see it. The disadvantage is that it requires extra computer software at both the server and for people looking at the information.

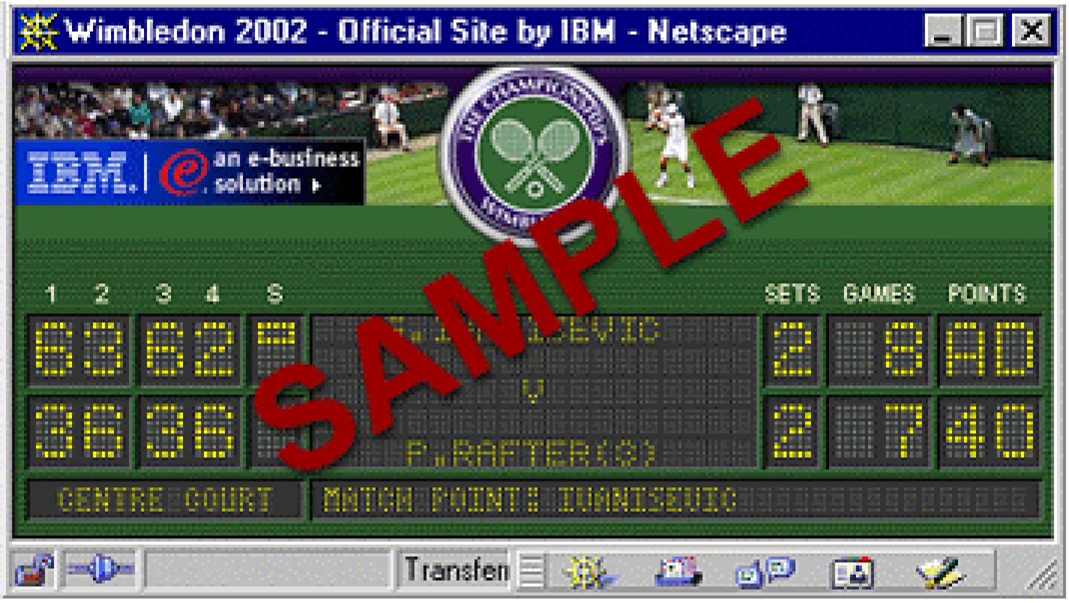
Now the picture looks like this;



We have added another computer to the system that can produce individual webpages based on the matches that the user wants to see (different colour webpages in the picture!) We could just put the scores up on as a piece of text on a webpage, but that isn't very interesting to look at.

To help the user feel like they are part of the action ("at Wimbledon") we could make the webpage look like the big scoreboards at Wimbledon. The real life "IBM Realtime Scoreboard" program looks like this;

Anyone for Tennis?



Over the course of the Wimbledon fortnight around 4.2 million of these scoreboards are used by visitors to the Wimbledon website – an average of around 30,000 during each day. What do you imagine the peak demand for this application is?

The highest number of simultaneous users last year was 169,000... Step 3 in the architecture process is to consider the hardware and software required to support what you're trying to achieve, but we'll leave it here!

Tell me your secrets?

Another significant part of the Wimbledon Championship website is the online shop (you saw a screenshot earlier). It allows visitors to order Wimbledon merchandise from tennis balls and towels to cups, books and memorabilia.

The customer specification for this facility probably looked something like the following;

> *We want to put our souvenir shop online so that people around the world can order items from us 24/7.*

As before, the first step is to gather the requirements;

- **What is in the product catalogue?**
- **Where are the customers? How many?**
- **What is the expected usage pattern?**
- **What does 24/7 mean?**

What sort of products are in the catalogue? We already talked about tennis balls and towels – what other ones? How many items are there? What kind of value are the items? (high priced items, low priced items, a mix?) Are they already on a computer system or would that have to be done too? (this would require lots of scanning pictures and manually entering information)

Where are the customers? Chances are that for this type of shop the customers will be worldwide.

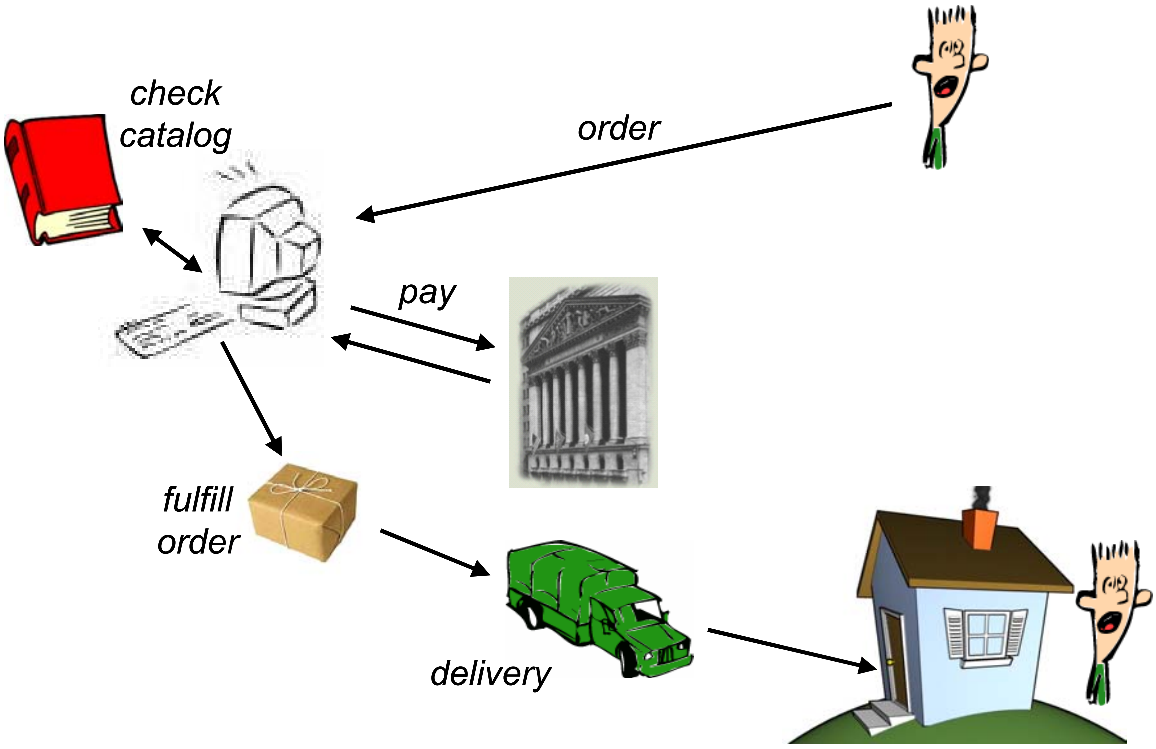
How many customers? For Amazon, their busiest day this Christmas say 2.1 million items ordered. (a single day). That's 24 items a second...

What is the expected usage pattern? All at once, spread out? Peaks during the day? (Whose day?!)

What does 24/7 mean? Is there a quiet period that can be used for maintenance / hardware replacement? It's expensive to have systems that can cope with heavy demand twenty four hours a day, every day.

Using the same kind of approach as just now you can quite quickly come up with the following system overview diagram;

Anyone for Tennis?



The customer looks at the online shop and orders some goods. The computer checks with the online catalogue to ensure things are in stock and if so it communicates with the bank to take payment for the goods that have been ordered.

Soon after that someone in a warehouse at Wimbledon packages up the goods that have been ordered and puts them in a pile ready for delivery by a delivery company to the customer's house

The challenges for this system include the following:

- How do you deliver the orders that you take – speedy and efficient delivery is as much a part of what people think of your site as whether the website is nice to look at.
- How do you take payment? Cash, credit card? How can a customer give you their credit card number over the Internet and be confident that it won't have been intercepted by unscrupulous persons on the way?

The second question is what we are going to consider now. It can be stated in the following way;

> *How can I give my pay for things over the Internet without my credit card number being found out by criminals as it is transferred between me and the website?*

Why is security important when doing things over the Internet? The answer is that it's a big old world, and it is easy to eavesdrop on transmissions between one computer and another. The offline analogy sending your credit card number through the post – how can you make sure that anyone who opens your post cannot take your credit card number and use it for themselves?

Regular transmission sends the sensitive information in plain form. Anyone watching traffic on the Internet could see your credit card info. You might be lucky, but do you really want to take that risk? Once they have your credit card number they can user it to order stuff themselves – free holiday!

Anyone for Tennis?

1234 5678 9012 3456

So how do you protect your credit card number when you send it to the website and the other end?

Computers can encrypt a message by applying a mathematical function to numbers that represent the message you want to send.

First of all, think of an equation that you want to use to encrypt the number and tell anyone you want to send the message to. Part of this function must include the ability for you to customize the way the equation works so that it is unique to you.

$$X = f(x,n) = 7(2nx + 75)$$

x is the number you want to encode, and n is a special number that you need to know to decrypt it.

As an example, take x=1234 and n=17 then work out X, the encrypted version (X=294217). To decrypt the encrypted number (ie message) you can simply invert the function and insert your personal number again.

$$x = f^{-1}(X,n) = \frac{1}{2n}({}^{X}\!/_{7} - 75)$$

The personal information, “n” is called the ‘key’, and is should be difficult to guess, so that the criminal (who might have found out the function) cannot work backwards from the encrypted number to get your credit card number.

Encrypting your message in this way is a like locking it in a cash box, then sending the cash box through the post to the recipient. Now no-one can read your message while it is in the post, which solved the first problem we had.

The problem now is that the person receiving your message (the website) cannot read the information without having the key to the cash box – which only you have...

How do you get the key to the person receiving your message? You can't send it over the Internet without locking it in another box (which would need a key) – it would be intercepted and could be used by the criminal to decode your credit card number. This problem of key distribution has plagued people wanting to send secret messages for 2000 years.

You could travel to the company and physically hand over the key, but that isn't really feasible as you might need to travel around the world to do so. In World War II both sides went to great lengths to distribute large books containing a sequence of keys that were used to encode and decode messages sent on particular days. Despite the immense cost this was the only way they could guarantee the secrecy of the messages they were sending. The problem was that every now and then the enemy would capture some of these books, and were able to read all the messages without any trouble...

A better way?

What if you could encrypt a piece of information in a way that did not require you to also send the information used to decrypt the information?

Image that instead of using a cash box to transmit your message you used a plain box with a padlock. **What if it was the type of padlock that you can snap shut without the key, but not open again?**

Now the person who you are sending the message to could send you a padlock that can only be opened with their key. You write your message, put it into the box and snap the padlock shut. Now the only person who can unlock the box and read the message is the person who sent you the padlock (you can't even read your own message any more!) You can send them the message by whatever route you like safe in the knowledge that only the intended recipient can read it.

What we have just described is called public key cryptography. There are two stages to sending the message – in the first you use a widely available piece of information (the padlock) to encrypt the message. This is the public key. The second stage requires a piece of secret information (the key that opens the padlock, or 'private key') to decrypt the message. A person who wanted to receive encrypted messages could send out millions of copies of the padlock that is opened by their key, and anyone who wanted to send them a message could use one to secure the box containing the message.

In computing terms both the public and private key are numbers. Clearly they are linked, but the trick is finding a mathematical function for which the private key cannot be easily determined from the public key.

Imagine that my private key consisted of two numbers, and that the public key was simply these two numbers multiplied together. If the public key is small, then you can easily work out what my private key was;

33	(3 * 11)
30	(2*3*5, so the numbers are either 5 and 6, 2 and 15 or 3 and 10)
323	(17 * 19)

Now what if my public key was 97845103?

This is the product of two four digit primes (9973 and 9811), and is approximately 10^8. It is a number that could be factorised using a computer in a fraction of a second, and so for real security much larger primes must be used.

The larger the amount of computing power you have access to, the easier it is to crack. If you had access to the computing resources of 100 million personal computers, then you could factorise a number the size of 10^130 in about 15 seconds. (That's 1 with 130 zeros after).

Because of this, for commercially sensitive information the public key is normally more than 10^300. It takes a fraction of a second to find the public key if you know the two primes that make up the private key, but would take around 1,000 years for your 100 million personal computers to work out the private key (the two primes) from the public key.

Of course if someone invented to factorise large numbers quickly then we would need to find another function, but mathematicians have been trying that for hundreds of years, and haven't made any progress so far.

Summary

We've discussed the following topics

- **Connecting the physical world to computer systems**
- **Behind the scenes on a global website**
- **Designing IT systems (it's not big, and it's not clever)**
- **Sending sensitive information over the Internet**

- **What skills do you need to work in IT?**

Hopefully this presentation has put across the idea that the most important skill for working in IT is nothing to do with IT itself, but is about being able to apply your experiences and skills to the problem at hand.

References

For further information on topics mentioned in this presentation, try the following – or send me an email: matt.roberts@uk.ibm.com

The Wimbledon Tennis Championships
http://www.wimbledon.org

IBM Wimbledon
http://www.ibm.com/wimbledon/

Hawk-Eye
http://www.hawkeyeinnovations.co.uk

Public Key Cryptography
The Code Book by Simon Singh
ISBN 1-85702-879-1