the lattice of value
designing products for self-growth

Alan Dix
alan@hcibook.com
http://www.hiraeth.com/alan/ebulletin/lattice-of-value/

background

In 1998 some colleagues and I were working on the business plan for aQtive, subsequently funded by 3i.

As we considered at our product range and marketing budget, we realised that traditional marketing was too expensive and we looked for ways in which our products could 'market themselves'.

One of the results of this was the 'lattice of value' a way of looking at products to both assess how well suited they are to self-growth and how they can be modified to improve their ability to grow.

This theme was followed up later within aQtive in the modelling and analysis of market ecology – modelling of the interactions between market groups and the potential for designing products to maximise growth through those interconnections.

co-value

The key to getting a product used (and bought!) is value. Do the users perceive that they will gain more value form it than it costs them – whether in effort or money?

To achieve self-growth we want users/customers to influence one another – because someone else uses product A I want to use product A or some associated product too. This is a form of co-value – I gain additional value from my use of product B because you use associated product B.

Although I didn't know it at the time there is a whole literature on the economics of this – network externalities and network effects [e.g. Brynjolfsson, 1996; Economides, 1996; Liebowitz , 1998; Weitzel, 2000]. Economists are interested in the phenomena as it means that free markets may not work to optimise production in the way that traditional theory suggests. From a design viewpoint however, we can take advantage of the effect to increase market growth.

One of the most obvious examples of this phenomenon is telephone use. The first user of the first telephone has no value from it (apart from it being a hi-tech ornament) as there is no one to call. The millionth telephone customer has far greater value as there are 999,999 existing telephone owners to talk to! Furthermore the first telephone user now has a more valuable product because others have bought telephones too.

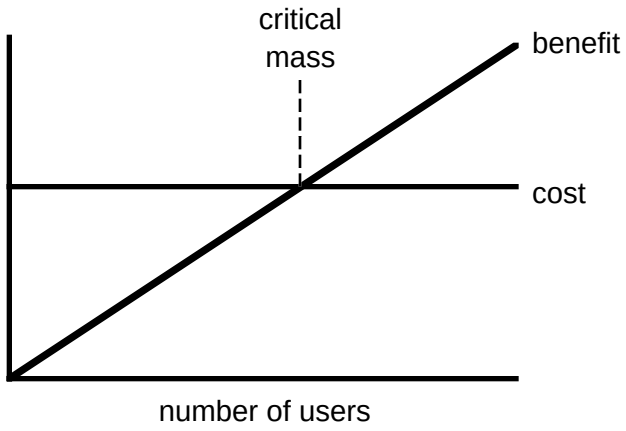

Figure 1. critical mass

This leads to two effects:

- **critical mass** – the first few users may have less value back from the product than the cost to them, so it is very difficult to get these first few users, on the other hand once the number of users exceeds critical mass it becomes easier and easier to sell the product (see the CSCW literature [Cockburn, 1993; Dix, 1997; Grudin, 1988])
- **self-growth** – those who have the product benefit if others also have it and are therefore likely to become evangelists for the product, promoting it to their colleagues and other potential customers

the lattice of value

The lattice of value captures the conditions for products to achieve self growth when there are two types of users and products.

It was developed thinking particularly of web-site developers and web users, but is not confined to these groups.

Suppose we have two products A and B. Product A is designed for web site owners and product B is a companion product for web site users. – for example, Adobe Acrobat Exchange to produce PDF documents and Adobe Acrobat Reader to view them.

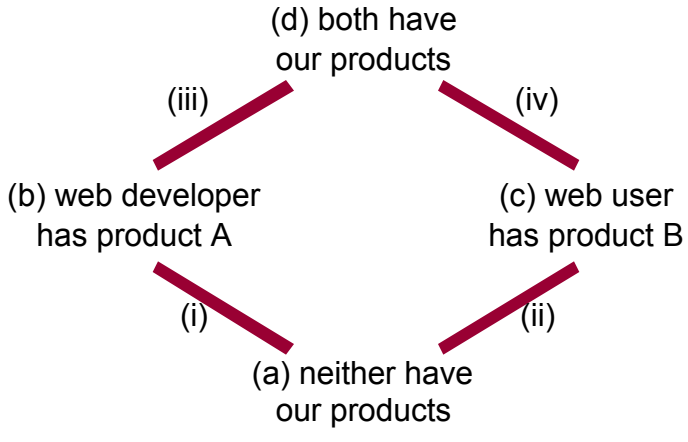

Figure 2. the lattice of value

We want these products to have the following value properties (figure 2):

(i) (b) is better than (a) – it is better for a web developer to use product A than not, even when there are no users of product B

(ii) (c) is better than (a) – it is better for a web user to use product B than not, even when there are no web sites using product B

(iii) (d) is better than (b) – if a web site uses product A then there is additional value for the end users if they have product B as well

(iv) (d) is better than (c) – if users have product B it is even better if the web site uses product A as well

Both (i) and (ii) are boot-strap properties, they get the product started. This avoids the critical mass problem – there is value even when there are no other users. Strictly only one is essential, but getting both is even better!

The other two properties (iii) and (iv) are the co-value properties ensuring growth.

The first of these (iii) means that web developers who use product A want users to also use product B, so become promoters of the product (producer push). For example, if you use Macromedia Shockwave on your site you encourage your web users to download Shockwave player.

The second (iv) means that users of product B actively want web sites to use product B, either explicitly through comments or suggestions to the web master or implicitly because the web developers knows their users want it – consumer pull.

designing for the lattice of value

Let's imagine we are thinking of producing a tool to help web developers create site maps. The tool will run on the developer's machine, crawl the web site and produce a site map that can be included in the site to help users navigate round it.

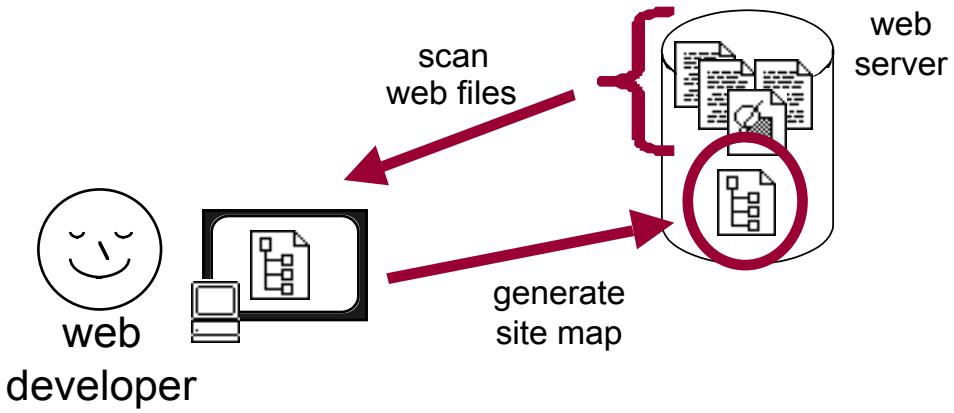

Figure 3. web developer site map tool

On its own this product only satisfies condition (i), assuming it is good, it is better for web site developers to have it, but there is no companion product to promote growth. Possibly if we have a 'powered by' logo on the site map other web developers may see it and adopt it, but we have no associated web-user product.

So, we look at some sort of client-side downloadable product that acts as a browser plug-in, and builds site maps of pages the user has visited. A kind of history that also keeps track of site structure.

Figure 4. web user site map application

This second product satisfies (ii), again assuming it is well designed, and so users may adopt it, but it has no relation to our developer product.

To get this interaction we add some features to the products. First of all we update the developer's tool so that it automatically uploads a site summary file whenever a site map is produced. This file includes information about changes to the site and possibly information about each page such as title, keywords, etc. We then make the end-user product look for this additional information and if it is there incorporate the information, giving the end-user an instant site map with information about changed pages, the ability to search the site by keyword etc.

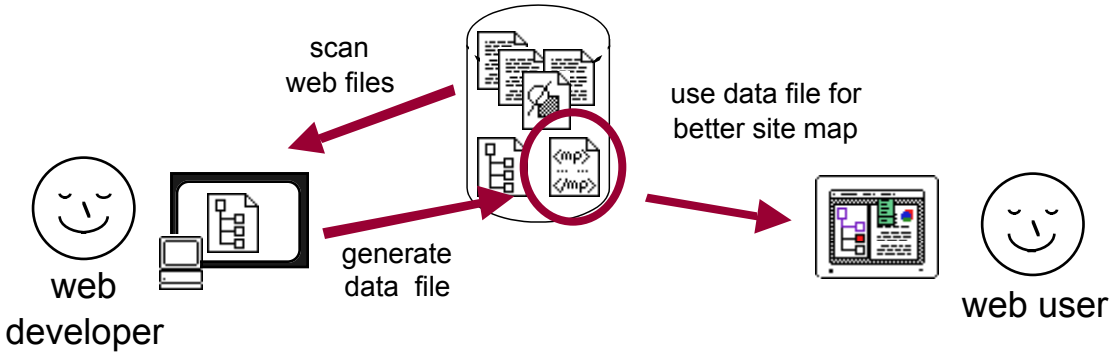

Figure 5. updated products

Let's look again at the co-value proposition ...

First of all (iii) – if you are a user of a site with the site map, you already have the site map, but it doesn't know what you have visited. By downloading the end-user application the value of the site map is augmented.

Similarly for (iv) – if a user has the end-user application then sites with the extra summary information are easier to use and have extra information about changed files etc. .

additional value issues

Conditions (i) and (ii) are also important for growth.

Suppose the product has property (iii), but not (ii) and a user encounters a site that uses product A. Although that site may become better if the user gets product B, other sites are not improved. So unless the gain is really big, or the site really important it won;t be worth the effort.

Similarly, if (iv) holds, but not (i) we would be asking the web developer to adopt a new product just to help an individual user.

Note that (i) and (ii) may actually be negative. For example, designing a web site to work well with a particular browser plug-in (for example, basing the site on Shockwave) which means increased value for those users, but makes the site less good for other users.

Also note that we need to be a little careful in looking at value for (iii) and (iv) as there are two kinds of value – value for the developer and value for the user. For the web these are often aligned – value for the producer means a better web experience and so value for the user. However, this is not always the case, for example, if we made the end-user site-map plug-in upload site usage data, which would add value for the developer, but not the user. Designing an appropriate lattice of value for a product means that we need to make sure that conditions (iii) and (iv) are true for **both** developer and user.

winning without the lattice

Products can achieve self growth without all the conditions holding, in particular (i) and (ii), but usually only if there is some alternative way of achieving critical mass.

One way is shear market force. For example, if Microsoft delivers a product free with the standard Windows installation generating a consumer pull (arguably the situation with IE's special features). Acrobat similarly benefited from Adobe's market presence and large PR effort.

Another way is to reduce costs on either producer or consumer side. For example, there is no value to having Acrobat reader if you never encounter a PDF file – condition (ii) fails. However, Acrobat reader is free so that even modest gains to the user of obtaining the reader when encountering an Acrobat document (condition (iii)) outweigh the costs of obtaining the reader (download).

Early on in Acrobat's adoption even this was not strong enough as slow modem speeds made downloads expensive in terms of time and for some telephone costs. However and alternative market group, documentation producers, had a different value balance as they could deliver Acrobat to end-users on the same CD ROM as the documentation itself. This lead to penetration of the reader into end-user groups and thus consumer pull to

use PDF documents on web sites. This shows that growth is not just about one or two market groups and products, but about whole webs of interconnections. This is the study of market ecology.

references

Online links to the papers below and other related links at:
http://www.hiraeth.com/alan/ebulletin/network-effects/

- Brynjolfsson, E. and C.F. Kemerer. (1996) Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market. *Management Science*, **42**(12):1627-1647, December 1996

- Andrew Cockburn , Harold Thimbleby (1993). Reducing user effort in collaboration support. *Proceedings of the international workshop on Intelligent user interfaces*, p. 215-218, January 1993, Orlando, Florida, United States

- A. Dix (1997). Challenges for Cooperative Work on the Web: An analytical approach. *Computer-Supported Cooperative Work: The Journal of Collaborative Computing*, **6**:135-156.

- Nicholas Economides (1996). The Economics of Networks. International *Journal of Industrial Organization*. October 1996,

- Grudin J. (1988) Why CSCW applications fail: problems in the design and evaluation of organisational interfaces. In: CSCW'88 Proceedings of the Conference on Computer-Supported Cooperative Work, ACM SIGCHI & SIGOIS, pp. 85–93.

- Stan Liebowitz and Steve Margolis (1998). Network Externalities (Effects). entry in *The New Palgraves Dictionary of Economics and the Law*, MacMillan.

- Tim Weitzel, Oliver Wendt, Falk von Westarp (2000). Reconsidering Network Effect Theory. In: *8th European Conference on Information Systems (ECIS 2000)*; Wien, Österreich.