

# THE ECOLOGY OF MONEY: SOFTWARE CHALLENGES

Hugh Barnard UK MSc [hugh.barnard@hughbarnard.org](mailto:hugh.barnard@hughbarnard.org) Independent Researcher

## Theme and Field

Field: Monetary Theory and Computer Science

Theme: Practice

## ABSTRACT

Difficulties, challenges and opportunities for computer systems that support radically reformed methods for the storage and exchange of value. This subject is investigated in the context of the partial replacement of conventional money (which seems to often cause so much damage) with new forms and expressions of value.

## INTRODUCTION

By way of Introduction, I'd like to draw together three strands that introduce the subject:

1. [1] George Santayana: "Those who cannot remember the past, are condemned to repeat it,"
2. The Baby and the Bathwater: Throwing out the baby with the bathwater...
3. [2] Thick and thin descriptions, value as a context free part of 'economics' and value as part of a 'culture'

These three are some of things that I think about, when thinking of humane finance and financial systems. To deal with each of them.

One lesson of the Santayana quotation is in attempting to create something new we create something that we have already visited and fall into the same ways and traps. Let's try what philosophers call a 'thought experiment', imagine we have the **same system of finance**, including credit with interest, complex financial instruments, tax havens, international flows and modes of payment **but it is run by 'the people' rather than by 'for-profit'**.

Firstly, it's clear that, unless the rest of the world is also run by 'reformers', there's no difference because outflows would 'punish' something that was run for human values rather than paper profit. That's not a criticism, it's an assessment of the state of the international culture associated with exchange. This is also the reasoning that appears in Simpol [3].

Secondly, the baby and the bathwater. First, an explanation of this image. The bathwater is dirty and we need to throw it away, but we should not discard items of value, simply because both these things are connected in a contingent way. In the same way, there are probably some

features in the current way we 'do' exchange and storage of value that are worth retaining and some are not.

And lastly tying these two together, the theory of thick and thin descriptions [2], which I shall deform slightly for my own purposes. Many economic theories, including some quite radical ones, treat exchange and storage of value as a mathematical, culture-free subject. This is one of the most relevant and serious complaints made by Schumacher about economics [4] in his original essay. But real finance is not like that, it's embedded in a culture and influences values, attitudes and (worse) decisions within a given culture. It has a 'thick' description, although, when we pretend to be scientific (or try and be neo-classical economists), we pretend that it doesn't.

### **CURRENT SYSTEMS OF FINANCE**

First of all two qualifications. I'll be talking mainly about western finance and the banking 'industry'. This is what I know best.

Secondly, more seriously, I haven't distinguished between, 'money', 'financial systems', 'financial instruments' and the banking 'industry'. My intuition, which is somewhat reinforced by the events of 2007 and 2008 is that they are strongly interconnected. There is no way to prove that, however.

So let's try and 'do some philosophy' and 'describe' this system. I would hope, at the end to have a set of attributes that give a thick description. The 'thin' description is simply 'storage and exchange of value'. That's all, according, to many views of economics that finance does.

- scale: large scale covers countries, continents and sometimes the world (reserve currencies)
- ownership: run (mainly) by non-accountable, for-profit institutions (no, governments are not really in control of it, anymore)
- complex: contains many currencies, many financial instruments, some complex
- arbitrageable: money can make more money
- not mapped to reality: most money and related instruments are fiat, they are not related to real resources. Exception are, for example, physical commodity contracts and futures
- not mapped to ecology and the state of the world: accounting for flow and activity rather than (for example) pollution and resource consumption
- fetish and emotion linked: money has cultural and emotional 'value', 'greed is good', 'recession is bad', '-growth- is good'
- (possibly) creates inequalities: does not redistribute sufficiency, in spite of many efforts by

central governments over many years

- dominates human decisions: money and numbers make decisions, rather than humans making decisions and then providing resources
- industrial rather than tool-based: uses many people and sophisticated machinery
- cultural (in the West anyway) we admire people who have a lot of numbers and those numbers do bring actual power over others
- a first class object in the computing sense, treated as if 'real' and subject to manipulation to produce other expressions of money
- although apparently abstract, lack, in many societies, causes real suffering

I'd also like to go further into a couple of financial instruments, just to demonstrate that they are linked to specific cultural values and ambitions.

### **Joint Stock**

The first is common stock and the joint stock company [5]. This evolved in the UK and was used to spread risk in ventures that involved large boats and long and risky voyages. If a vessel and cargo (and crew, incidentally) was lost, then the loss was shared amongst the stockholders.

To some extent, I find this virtuous, it's a way (if kept simple) for people to work together on 'large' projects (which I believe humanity may always need).

### **Physical Futures**

Physical futures [6] are contracts for delivery of commodities at some point in the future. Though first recorded in Japan, the main development was in the US in Chicago to deal with uncertainties in the price and delivery of grain and agricultural commodities. They smoothed out prices for both the producer and consumer.

## Summary

Both these instruments have the following things embedded into them:

- Non Local Sourcing: They deal with bringing 'stuff' from 'far away'
- Mitigating large scale risk: Large scale things are 'risky' (the risky part needs looking into)
- Later, evolution into pure financial objects: Interest rate futures, modern stock and exchanges

To some extent, we 'need' these instruments because we have decided to live our life 'this way' or, more likely, we haven't really examined any of this closely and have just taken it as a priori (Schumacher's complaint and that of other non-traditional economists).

It also notable that some of the instruments that have probably caused the most trouble recently, such as Credit Default Swaps are also supposed to mitigate or transfer risk, in this case risk to other numbers, not to anything that we would consider 'real'.

## FUTURE SYSTEMS OF FINANCE

I see what's needed now as being transitional, the challenge is to live, to a greater extent, without units of account. It doesn't even seem logical that anyone should be hungry or ill, just for the lack of pieces of printed paper or numbers in computer system. Why would we want to live like that? Especially when that system is large, far away and is making a better life for others by producing more numbers.

Any future financial systems will also be embedded in a culture and contain cultural values, but the numbers and calculations must not dominate or deform the the culture. It must have it's own thick description or, more probably, it will grow one, as it evolves.

That said, and moment of anger over, the kind of systems I am thinking about will have the following qualities:

- small scale and low power: can be run, managed and be effective in small communities, a tool rather than machinery, in Schumacher's sense [4]
- low technology: run on 'old' computers and can be modified by people with medium computer skills
- convenient: mobile phones and cards make things easier to administer and lower any costs in 'national' currency [8]
- very low transaction cost, meaning that very small transactions are feasible, this fits with the microfinance [7] model too
- capable of accounting for multiple currencies for multiple representations and uses (see

later, for example non-renewable resources shouldn't be treated in the same way as renewable)

- open and open-source[9] : a group that doesn't like a particular 'system' should be able to easily walk away and start another one (and indeed this may be the key to evolution)
- simple: there are no (complex) financial derivatives [10] and little opportunity (apart from minor administration) to 'make money with money'. The money space has become 'flattened' (again)
- mapped to the world: units of account have a relationship to the world (for example, energy, crops or commodity basket backed currencies)
- part of a loosely linked ecology of community exchange but not strongly meshed (as at present)
- transparent and responsive: governance and ownership is always direct by the community, and often democratic
- loosely coupled: communities inland may want to trade fish with communities nearer water, for example
- post-scarcity: people will not suffer for the lack of any of it and indeed for many transactions, it will disappear

### **Consequences for Computer Systems**

- The 'bank' can run on an old PC or a small low processing power PC (I'm thinking of the new ones that draw 20 watts, for example) This is not yet within range for solar power though...
- A great deal of the user interface is via wireless (either wifi or mobile phone or a mixture), then there is no wired infrastructure (this tends to get stolen in developing societies or break too!)
- All the software stack (that's all the components needed to run the 'bank') are open source [9], operating system, programming language, database, network interfaces
- The programming language is fairly low technology, doesn't require huge skill levels
- The programming language is multi-platform, so that the software can, for example, run on Windows (not open source but widely available), Linux and Mac OSX (Java[11], Perl [12] and Ruby [13] are examples)
- The application part can be translated and 'reskinned' into many human languages without touching the application core

- The system can communicate with other 'banks' (preferably even those using another type of community oriented software) without involving commercial intermediaries (although there's a question in my mind about non-profit clearing houses)
- There are facilities for mapping and issuing currencies against a chosen measure of 'real-world' reserve commodity (energy, food, commodity baskets)
- Low maintenance, nearly all 'normal' operation is automated, the need for administration is low (and thus the administration 'cost' is low)
- Can be installed and run easily, by a non-professional, this implies good documentation in the local language and perhaps some support from the software developers

### **ARE WE THERE YET?**

In the past four or five years, there's been a lot of software development in this area, but I believe that there's still quite a long way to go:

- Systems will be 'simpler' and small-scale, this is good news for access to and community ownership of such systems
- 'Easy' for software developers is 'hard' for potential users. Users (anyway) cover a wide range of skill levels and levels of sophistication
- Many developments still need various bits of supporting documentation
- Most systems are single currency or single 'unit of account'
- Many developments are specifically for one view of currency (usually LETS [14], or modified versions of mutual social credit systems). Expect experiment and evolution.
- Not much attention has been paid to secure systems, these systems will eventually be targets for intrusion
- In most of the developed systems, they are often web-only, whereas conventional money has many more interfaces. Commercial transfer in Africa is often mobile-phonned based [8], for example.
- Some of the systems cannot be localised[15] into another language without touching the application core
- Many systems are single platform, for example those that use the Microsoft Access database. Small communities cannot usually buy into the costs associated with recent versions of Windows
- There's a tension between the complexity required and the ease of use and maintenance.

This will always remain somewhat problematic.

In my mind, I see my 'bank' as an old computer in a box nailed to a tree or telegraph pole. It occupies a very subordinate position in society and no bankers are gathered around to worship it.

Technically, it has undergone an alignment with systematic ecological thinking (accounting for non renewables, being 'careful' with energy, reducing waste) and culturally, we no longer make a religion of these numbers. Both the thin and thick descriptions have changed quite substantially.

### **Some Software Resources**

There are current and incomplete lists of software at:

<http://uo.twentiethcentury.com/index.php/FacultyEconometrics>

<http://www.letslinkuk.net/software/>

Some software depends on Microsoft Access and some are not free in the financial sense.

Most of this software is LETS oriented, with perhaps the exception of Cyclos [17], Ripplepay [18] and (maybe) my own Cclite [19].

### **SOME AFTERTHOUGHTS**

Well, no-one can see the future, but we can rehearse it and we can make it, to some extent. So these are some of the things I think about, in a set of progressive phases:

- Money is used to account (again), not to dominate or profit or produce 'more' money
- Different goods, services and societies need different monies or units of account, especially the distinction between renewable and non-renewable, for example
- We still need 'something' that diminishes the complexity of counter-trade and barter. That is probably something that is somewhat like 'money'
- Some parts of the modern age are very attractive (modern medicine, for example) and still need significant infrastructure, we still need joint venture
- Money should disappear for food, shelter and (a certain amount of!) energy

### **REFERENCES**

Note: All of these are online where possible, this is to give readers access to the supporting information without using a substantial (or university) library. All you need is an internet cafe and this list! This reduces the appearance of 'scholarship' but I hope it makes it more useful!

- [1] George Santayana (2009), [http://en.wikipedia.org/wiki/George\\_Santayana](http://en.wikipedia.org/wiki/George_Santayana)
- [2] Ryle and Geertz (2009), [http://en.wikipedia.org/wiki/Thick\\_description](http://en.wikipedia.org/wiki/Thick_description)
- [3] Simpol (2009), [http://www.simpol.org.uk/Pages/Welcome\\_FS.htm](http://www.simpol.org.uk/Pages/Welcome_FS.htm)
- [4] E. F Schumacher (2009), [http://www.schumachersociety.org/buddhist\\_economics/english.html](http://www.schumachersociety.org/buddhist_economics/english.html)
- [5] Joint Stock Company (2009), [http://en.wikipedia.org/wiki/Joint\\_stock\\_company](http://en.wikipedia.org/wiki/Joint_stock_company)
- [6] Futures (2009), [http://en.wikipedia.org/wiki/Futures\\_contract](http://en.wikipedia.org/wiki/Futures_contract)
- [7] Microfinance (2009), <http://en.wikipedia.org/wiki/Microfinance>
- [8] Use of mobiles for finance in Africa (2009), <http://news.bbc.co.uk/1/hi/world/africa/6510165.stm>
- [9] Open Source Software (2009), [http://en.wikipedia.org/wiki/Free\\_and\\_open\\_source\\_software](http://en.wikipedia.org/wiki/Free_and_open_source_software)
- [10] Financial Derivatives (2009), [http://en.wikipedia.org/wiki/Derivative\\_\(finance\)](http://en.wikipedia.org/wiki/Derivative_(finance))
- [11] Java Language (2009), [http://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Java_(programming_language))
- [12] Perl Language (2009), [http://en.wikipedia.org/wiki/Perl\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Perl_(programming_language))
- [13] Ruby Language (2009), [http://en.wikipedia.org/wiki/Ruby\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Ruby_(programming_language))
- [14] LETS (2009), [http://en.wikipedia.org/wiki/Local\\_Exchange\\_Trading\\_Systems](http://en.wikipedia.org/wiki/Local_Exchange_Trading_Systems)
- [15] Localisation (2009), [http://en.wikipedia.org/wiki/Language\\_localisation](http://en.wikipedia.org/wiki/Language_localisation)
- [16] Arbitrage (2009), <http://en.wikipedia.org/wiki/Arbitrage>
- [17] Cyclos Project (2009), <http://project.cyclos.org/>
- [18] Ripplepay (2009) <https://ripplepay.com/>
- [19] Cclite (2009), <http://www.hughbarnard.org/?q=node/3>