# The theory of deferred action: Designing organisations and systems for complexity

Nandish V Patel
Brunel Business School
Brunel University
Uxbridge UB8 3PH
UK

Nandish.patel@brunel.ac.uk
01895 265295

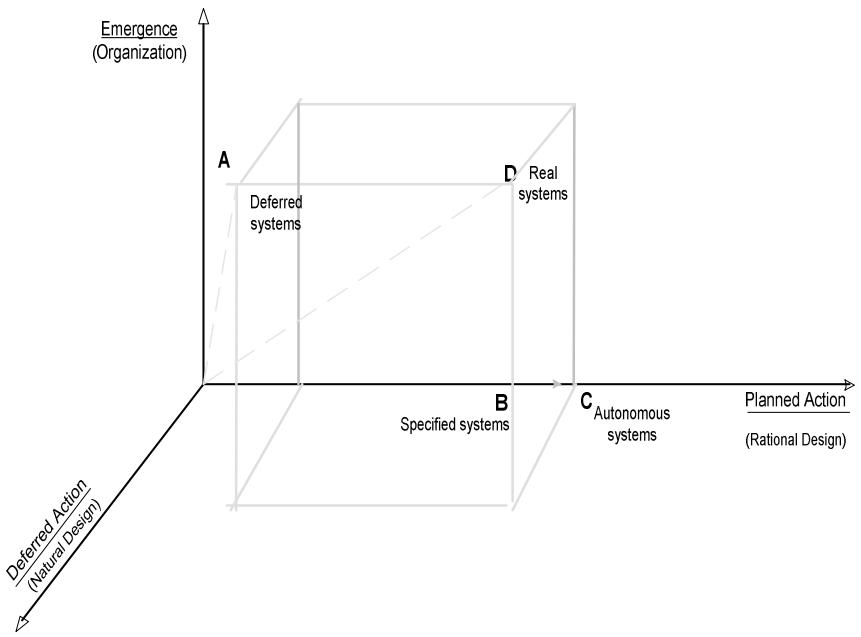
**Who benefits and what kinds of design problems does it address?**
Deferred design will benefit IT systems designers, business organisation designers and researchers. Deferred design addresses design problems where there is uncertainty about the problem domain. Where 'users' find it difficult to specify the information they want, making it difficult for designers to making design decisions based on an exact requirements specification.

## The Theory of Deferred Action

Though rational planning is necessary, in the context of emergence it is insufficient as the sole design dimension. Its scope is limited because agents modify their behaviour in the environment resulting in emergent organisation. The theory of deferred action provides understanding of systemic emergence to design complex adaptive systems (Patel, 2006). It is a generic artefact design theory for emergent organisation. In Gregor's (2006) terms, it is a 'theory for action and design' and therefore it informs design practice. Nomothetically, it explains and suggests effective models for organisation design, IT systems design, IS and KMS design, where emergence is a critical design factor. Here we invoke it to improve emergent IT artefacts design.

Three dimensions for designing emergent IT artefacts are postulated: planned action, emergence and deferred action, and their interrelationship constitute rational design of emergent IT artefacts, as depicted in Figure 1. The theory assumes business organisations *rationally* determine goals and rationally plan to attain to realise them. A plan is any artefact whose purpose is to construct the future such as strategic business plans or new systems design. However, the theory assumes *actual* organisational behaviour results in emergent organisation. Therefore, since rational behaviour is tempered by emergent behaviour the latter needs to be catered *actively* in the rational plan. A further assumption is that actuality is emergent and takes precedence over central plans but agents actions are constrained by the plan. Therefore, plans accommodate actuality but the teleological purpose of the system should not be deflected by the emergence.

*Figure 1. Deferred Action Design Dimensions*




**Planned action**

Planned action, boundedly rational design, looks at future states of systems, designing new systems and enhancing existing systems. It develops new systems futures drawing on existing knowledgebases. The innovation of a new IS draws on existing knowledgebases for developing IT systems, such as IS methodologies and design languages like UML. When planned action is not affected by emergence systems can be specified, as depicted at point B in Figure 1. These are called specified systems.

Planned action is undertaken centrally. It may be IS plans or management strategies. It is action prescribed by design and enacted *regardless* of actuality. For example, a three-year strategic plan or formal systems design for ERP systems. Planned action characterises organised action exclusively as rational act. It is useful for design problems that can be predetermined and well-structured and for solutions that can be predetermined requiring explicit and declarative knowledge. It assumes stable organisation structure and processes and negates emergence. Planned action is necessary but not sufficient for designing CAS.

**Emergence**

Emergence is the patterns that arise through interactions of agents, interactions between agents and IT artefacts, and agents' responses to environment. *Emergence is a becoming aspect of design*. It affects design processes and the designed systems. Agents act locally in emergence situations. So, emergence requires present, contextual, and situational aspects to be factored into design.

To design CAS, planned action prescriptions need to cater for emergence. When planned action is affected by emergence systems cannot be completely specified. It is necessary to relate by synthesis planned action and emergence to design emergent IT artefacts, as depicted by points A and D in Figure 1. Planned action and emergence are related design dimensions when designing for emergent organisation.

**Deferred action**

Deferred action is the synthetic outcome of relating planned action and emergence for designing CAS. Agents undertake deferred action, within planned action, but their action is determined by and enacted in the emergent context. Thus adaptableness and self-organisation, characteristic of CAS, are facilitated as deferred action to design CAS. Deferred action is necessary to design successful CAS.

Deferred action reflects emergence, space (location), and time in planned action. It contextualizes planned action in emergent situations. Since emergence is unpredictable agents should be enabled to respond to it in particular organizational situations. Deferred action enables agents to modify an IS within the context of its use. So, systems at points A and D in Figure 1 should provide actors with deferred action capability. The IS product is conceptualized as continuous design and development process, rather than a time-bound, predetermined product.

The interrelationships among these design dimensions are detailed in Table 1 and they model designed systems in emergent *actuality*. Actuality is never sympathetic to plans. Plans are subject to systemic emergence and require an adequate embodied and situational response. In rationally designed CAS this response is deferred action.

*Table 1. Design Dimensions for Designing Complex Adaptive Systems*

| **Design Dimensions** | **Description** |
|---|---|
| Planned action | Rational planning is necessary to set and achieve organisational goals, to build goal-oriented structures and processes. |
| Emergence | Agents' local responses to the environment create emergent situations. Emergence requires systems design and organisation design to be *continuous*. |
| Deferred action | Deferred action takes place within planned action in response to emergent locale. It synthesises planned action and emergence. |
| Synthesis of these constructs results in four system types: deferred systems (point A), specified systems (point B), autonomous systems (point C), and real systems (point D) in Figure 1. These types are also generic design types, systems types and organisation types. | |

To illustrate, Google's organisation has the three design dimensions. Google has an IT infrastructure (planned action) that is 'built to build', providing the flexibility needed in emergent context. It is designed to enable further building by expansion and adaptation to market needs (emergence). Google executives realise that they are not best placed to know the emergence, so they actively enable employees to take action when they consider it appropriate (deferred action). Employees are given 10% of their time for creative work. Thus a Google employee blogger reveals how easy it was for him to write software code and have it implemented in Google's gmail application because he disliked a certain aspect of it. Google's organisation is a deferred organisation and its IT systems are deferred systems, as depicted in Figure 1.

**APPLICATIONS OF DEFERRED ACTION BASED DESIGN**

The theory of deferred action and the principle of deferred design decisions are used and applied by researchers in research systems and applied business systems. Various types of knowledge work

benefit from the deferred analysis including legal arbitration and organisational learning. This work tends to be of the semi-structured problem type. Two exemplars are given. Both these systems are in the domain of knowledge management.

E-Arbitraton-T system

Elliman and Eatock (2005) developed the online E-Arbitraton-T system capable of handling workflow for *any* legal arbitration case, thus meeting the emergence criteria. The project aimed to develop an online system for Europen SMEs seeking fair dispute resolution in an international forum. The system would be used by many different organizations offering arbitration services but the cost of adapting E-Arbitratio-T to local priorities, including emergent factors, had to be kept low as some organizations had low case loads making high cost unjustifiable. Elliman and Eatock applied the deferred analysis, particularly the deferred design decisions principle, to manage the open and changing system requirements, making their system an open system. This enabled users to make design choices rather than the system developer.

CoFIND system

The deferred action construct is reflected in deferred learning systems. Dron (2005) invokes deferred systems to design systems that have 'emergent structure', allowing the system to have changing functionality. He developed a self-organised e-learning web-based system called CoFIND. Self-organisation in CoFIND results in emergent structure which the system needs to reflect. It is not designed from requirements but takes shape in response to the actions of the people that use it.
What is the related, independent, development in IT?

#### EXPLAINING SYSTEMS PRACTICE AS DEFERRED ACTION

There is much independent invention of deferred technology shown in Table 4. These independent inventions serve to cater for emergence in digital applications of information and communications technologies. It is because fixed technological functions are inadequate in emergent situations that the deferred technology is invented. The function of the deferred technology takes form in actual contexts, contexts which could not be pre-determined by designers and developers.

*Table 4. Deferred Technology Implementations*

| Term | Description |
|---|---|
| Deferred-action-list; deferred-action function | Used in emacs-development. |
| Deferred Execution Custom Actions | Used in scripts for Windows installer. |
| Deferred Procedure Calls | Microsoft uses DPCs to manage hardware interrupts At micro-processor level: Microsoft's response to this problem is to use Deferred Procedure Calls (DPCs). http://www.nematron.com/HyperKernel/index.shtml |
| Client side deferred action with multiple MAPI profiles | This is a patent at: http://www.patentalert.com/docs/000/ Z00002860.shtml |
| Java deferred classes | Used in the Java computer programming language. |

### *Open Source Software as Deferred Systems*

Open source software is well explained by the theory of deferred action. Open source software is a complex adaptive system that is a deferred system. It has a planning core that determines the direction in which open source software will develop. This is the planned action element of the system. But the actual problems addressed by open source software are determined locally

(emergence) by individual software coders (deferred action). This is the emergent aspect of the system. The Linux operating system is an exemplar.

Extant information systems and information technology can be mapped in terms of the four design types, shown in Table 5.

*Table 5. Classification of extant systems in terms of deferred action*

| Deferred System | Real System | Autonomous System | Specified Systems |
|---|---|---|---|
| World Wide Web<br>CoFIND<br>ViPre<br>Dallas<br>Capital | Internet<br>Semantic Web<br>Air traffic control<br>Modern military systems | Intelligent agents | Payroll<br>Sales<br>Product databases |