

# DEFENDING ESSENTIAL PROCESSES

A. Alderson, A. Salter, H. Shah, B. Sharp

*School of Computing, Staffordshire University,*

*Beaconside, Stafford, England, ST18 0DG*

*Email: A.Alderson@staff.ac.uk; A.M.Salter@staffs.ac.uk; H.Shah@staffs.ac.uk; B.Sharp@staffs.ac.uk*

Keywords: Processes, Norms, Process design, Defensive norms, Sanctions.

Abstract: The essential purpose of a program makes up only a small part of the overall task. All of the complications in the program come from addressing what can go wrong. Where the essential business processes remain stable, close examination shows complex defensive mechanisms which change as new threats to the business develop. Norms derive from modelling social behaviour but are not absolute expressions of what will happen, people may act counter to the behaviour described in the norm. Many norms in business are concerned with defending against erroneous or illegal behaviour of staff and third parties. This paper uses examples to illustrate the development of defensive norms and how these norms may be used in designing processes. Essential business processes cannot be improved by adding defensive norms but processes are usually more effective where security norms are implemented preventing the breaking of norms.

## 1 INTRODUCTION

All programmers are aware that coding the essential purpose of a program is usually only a small part of the overall task. All of the complications come from addressing what can go wrong. The concept of ‘defensive programming’ was introduced to protect pieces of code from the effects of their environment and mistakes in implementation. A close examination of business processes shows complex defensive mechanisms, many of which we take for granted, which have developed over millennia. It is evident that the essence of business processes (the essential process) is stable but the defensive mechanisms change as new threats develop. There is clearly value in differentiating the essential process from its defensive additions.

Norms derive from modeling social behavior. They may be distinguished as perceptual (ontological - acknowledging existence, naming), evaluative (axiological - making value judgments), cognitive (epistemic - adopting a degree of belief) and behavioral (deontic - being disposed to act in some way). All of these norm types are relevant to the creation of business systems, but in this paper we focus on behavioral norms.

From Von Wright (1963) we have the elements of a norm that we interpret as:

- The authority for issuing the norm (e.g. a retail store).
- The subject governed by the norm (e.g. a customer).
- The circumstance in which the norm is relevant (e.g. the customer wishes to buy something).
- The occasion in space and time during which the norm is valid (e.g. whenever the items are available for sale).
- The deontic attitude (e.g. obliged, permitted, forbidden).
- The action prescribed (e.g. to pay for those items).

Norms are not absolute expressions of what will happen. People may act counter to the deontic attitude. Such behavior may be detrimental, neutral or beneficial from the viewpoint of the norm authority; advances often come about by questioning the ‘norm’. Defensive norms are a response to detrimental behavior and are usually associated with sanctions. A sanction can be a penalty for disobedience (a fine for speeding) or a reward for obedience (a bonus for meeting targets).

In system operation, norms determine whether and when a certain activity takes place; they also determine the behavior of the system in performing certain tasks. For instance, in a particular retail sales process a sales assistant must determine whether the customer has a customer profile card. If not, they must offer the customer the opportunity to register

for one; the customer can refuse. Such business rules can be collected and analyzed.

It is apparent that many of the norms of business are concerned with defending against erroneous or illegal actions of staff, customers and third parties. The creation of successful business organizations depends on understanding and dealing with non-compliance with the deontic attitudes of norms.

This paper uses examples to present the concept of defensive norms, differentiates classes of defensive norms, considers how the concept can be applied to business process design, and elucidates the relationship between norms and requirements.

## 2 ILLUSTRATIVE EXAMPLE

We can illustrate the development of defensive norms in relation to trade.

Merchant and Retail Laws have developed to regulate trade. Originally these norms were agreed generally (if not specifically) between merchants and enforced by mutual agreement through bodies such as the Courts of Pie Powder. These Courts of Pied Poudre, or dusty feet, traveled from market to market ruling on disputes between merchants who walked to the markets. These norms have been adopted and formalized by society as law. That penalties exist for breaking those laws indicates they are not always obeyed. It is still for businesses to defend themselves against breaches.

The essential process of trade is that:

- The supplier is obliged to deliver the requested goods.
- The customer is obliged to pay for the delivered goods.

Consider the subject of invoices. An invoice represents a request for payment for services or goods. A business might have the behavioral norm:

**Norm 1:** The business shall pay invoices when they are received.

Experience will teach that this is naive. Fraudsters send invoices for goods or services not delivered. A business using Norm 1 will be defrauded. For example, many firms pay false invoices for entries in international fax directories.

So, businesses assert their intention not to be defrauded by requiring that goods and services charged for have indeed been delivered. This leads to:

**Norm 2:** Pay invoices when they are received and proof exists that delivery has been made.

Again a fraudster may deliver some unwanted service or goods and demand payment, which would be forthcoming under Norm 2, (for example, paying for your windshield to be cleaned at traffic lights).

So, businesses assert their intention only to pay for goods and services ordered by requiring evidence of a valid purchase order. This leads to:

**Norm 3:** Pay invoices when they are received providing the goods and services have been delivered and were ordered. That is, the invoice, purchase order and delivery note all match.

A fraudster may deliver goods or services that are deficient. This might not be obvious through inspection. A period of use may be required. Demanding warranties is one defense, but involves paying over money. Subsequent attempts to invoke the warranty may be ignored. Action at law would then be necessary to reclaim money paid. This is expensive, so paying after some acceptance period would be wiser. This leads to the norm:

**Norm 4:** Pay invoices when they are received providing the goods and services have been delivered and were ordered, and they have been accepted (shown to be fit for purpose).

At the point of developing Norm 4 the company may recognize the advantage of delayed payment in improving its cash flow, again through what it sees as defensive norms. Although, its creditors will see them as offensive! We may have a norm such as:

**Norm 5:** Put each valid invoice in a queue and pay only the first 20, say, each week. This is an example taken from a real business.

We may have norms taking advantage of the relative strength of the business, as with:

**Norm 6:** Delay payments to a small business that relies on us for its orders.

This, of course, puts small businesses in financial difficulties. This may lead to the norm:

**Norm 7:** Delay payments to a small business that relies on us for its orders, and if it is in financial difficulties, do not pay at all. Wait until the receiver or administrator demands payment, then delay by negotiating.

This example shows how defensive norms add spiraling complexity to a very simple essential process. Defensive norms require that we introduce new concepts (and artifacts representing those concepts) and then introduce norms applying to the concepts as we build defensive norm sets, and so on. From our examples we have:

- Norm 1 - invoice;
- Norm 2 - evidence of delivery;
- Norm 3 - purchase order and delivery note;
- Norm 4 - warranty or acceptance period;
- Norm 5 - control cash flow;
- Norm 6 - relative business strength.

Clearly a business will seek the most effective concept or artifact from amongst the possibilities, having regard for risk and cost. This is central to the design of business processes. Note that new artifacts and concepts introduce new domains for the information systems of the business to cope with.

### 3 PENALTY, SECURITY AND EVIDENCE

If someone walks into a store and takes money from the till they can be punished for theft. Of course, if the till is kept locked the theft is more difficult. Here we have examples of the two kinds of defensive norms: penalty norms and security norms. Penalty norms are after the fact; a norm has been broken and a penalty is administered (e.g. criminal law). Security norms are before the fact; they aim to prevent norms being broken (e.g. passwords, door keys, encryption).

Penalty norms can only be invoked where there is evidence of breach. This requires the gathering of evidence during the process in case anyone should be shown to have defaulted after the fact. Security norms require some data to be provided and/or gathered before the fact that can be used as a means of identification or proof of right to the goods or services. Software engineers may recognize the similarity of security and penalty norms to pre-conditions and post-conditions in formal descriptions of programs.

We see that defensive norms have or are related to data gathering and retention activities. With defensive norms there is also a need to collect evidence to show compliance – the converse to collecting evidence of non-compliance (e.g. a customer may save receipts).

Defensive norms are often used as a means of quality control. However, defensive norms cannot increase the quality of the product above that possible by virtue of the essential process employed. Defensive norms can only prevent quality falling below some level. Defensive norms in quality control place an obligation to make quality measurements of some kind. To ensure that they are made, there will be a quality assurance function. And if the check is not made? Juvenal (AD c.55 - c.130,

Satires) recognized the problem in the first century: “*Sed quis custodiet ipsos custodes?*”.

### 4 DESIGNING PROCESSES USING NORMS

Software programmers are trained to consider the conditions under which pieces of code are to be executed, and to consider what action to take if those conditions do not hold. Using normative descriptions of processes we can ask business process designers to consider how to proceed if a deontic attitude ‘obliged’ or ‘prohibited’ is breached, or how to deal with alternatives if the deontic attitude is ‘permitted’.

Consider again retail trade. The retailer is obliged to provide the goods (of appropriate quality) in return for a consideration. The customer is obliged to pay. If the customer returns goods (with proof that they are not of appropriate quality) then the retailer is obliged to refund payment.

What if a thief steals goods then returns them to obtain payment (breaching the obligation to pay). The retailer may introduce the defensive norm: *if a customer returns goods then proof of purchase is required for the goods*. (The law requires the retailer to provide customers with proof of purchase - a receipt.)

What if the thief finds a receipt, steals the indicated goods and returns them to obtain a refund. The retailer may have the defensive norm: *if a customer returns goods with proof of purchase then do not give a cash refund – give a refund against a valid credit card or give a credit note*. This deters thieves whose motivation is to obtain cash.

What if the thief finds a receipt, steals the indicated goods and returns them to obtain a refund repeatedly. The retailer may have the defensive norm: *if a customer returns goods with proof of purchase give a refund and cross the item off the receipt*.

Note that the creation of such defensive norms requires great knowledge of the business environment. The manner in which various deontic attitudes of particular norms may be breached (accidentally or deliberately) requires insight and experience in that business. Construction of such norm hierarchies is not a matter for technicians. Business expertise is essential.

### 5 NORMS AND REQUIREMENTS

We would expect from their place in defining processes that norms would be related to

requirements and indeed there is a close correspondence.

We can express functional requirements in terms very similar to those used by Von Wright for behavioral norms. We have:

- The authority for the requirement – this could be the viewpoint or sponsor of the requirement. (e.g. a farmer)
- The subject governed by the requirement – on whom or on what system or component the requirement is placed. (e.g. requires a drilling machine)
- The circumstance in which the requirement is relevant – this may be a condition governing the validity of the implementation of the requirement or a utility function relating the degree to which a requirement is met to the cost of its implementation. (e.g. 4 meters deep and 60 centimeters in circumference within one hour)
- The occasion in space and time during which the requirement is relevant – it is frequently important to know the context of a requirement. (e.g. in stony ground)
- The deontic attitude (e.g. shall).
- The action or function required (e.g. dig a hole for irrigation)

This clear relationship prompts questions about the relevance of requirements engineering and system engineering techniques to norms in the design of systems. We might express the relationship by saying that requirements are norms for system designers and that norms are requirements on business processes. Requirements may be said to govern the implementation of a system, whereas norms govern its operation. Indeed where requirements cannot be automated, it is natural to express them as rules to be followed by the users.

## 6 CONCLUSION

The concept of defensive norms provides a useful technique to understand and design business processes. Consideration of the breaking of obligations can lead to the development of business systems that are robust and mission safe. It focuses attention on the purpose of system elements and aids cost-benefit analyses.

It is worth reiterating that an essential business process cannot be improved by adding defensive norms. Also processes are usually more effective, but possibly more costly, where security norms are implemented preventing breaking of norms.

Given the complexity of our world, there is not always a clear-cut distinction between essential and defensive processes. This does not negate the value of considering defensive norms in understanding business processes. A customer profile card in retail businesses may be seen as an essential business process for creating customer loyalty or as a defensive process limiting the impact of competition. Businesses and their customers may take different views of the same activity. For example, implementing a body search on customer leaving a retail store would have a very negative impact, but a body search on boarding an aircraft may be very reassuring.

We can see that the complexity of business processes is due to the necessity for defensive norms. The ideal business process is one in which the breach of the deontic attitudes of its constituent norms can be ignored. The greater the extent to which this is possible the better. We may take this as an indication of process quality. Breach of deontic attitude occurs when the person in breach is able to improve his or her own position to the detriment of others. Clearly processes where either party breaching the rules reduces mutual benefit are the best.

Work currently underway is considering the equivalence of non-functional requirements and non-behavioral norms. We are also studying in detail how system engineering models particularly that of Alderson *et al.* (1997) may be applied to norms.

## REFERENCES

- Alderson A., Arnold S., Brook P., Jackson K and Stevens R. (1997), An Improved Systems Engineering Model, *International Conference & Workshop on Engineering of Computer Based Systems (ECBS '97)*, Monterey, California, USA, 24-28 March 1997, IEEE Computer Society Press, 427-434.
- Peirce C.S. (1931-1958) Hartshorne C. & Weiss P. (eds.), *Collected papers of C.S. Peirce*, Harvard University Press, Cambridge, Massachusetts, USA.
- Von Wright G.H. (1963), *Norms and Action – Logical Enquiry*, Routledge and Kegan Paul: New York, USA.