

## Association for Information Systems AIS Electronic Library (AISeL)

---

AMCIS 2000 Proceedings

Americas Conference on Information Systems  
(AMCIS)

---

2000

# Information Systems Design Under a Different Light

Paulo Rupino da Cuhna  
*Universidade de Coimbra*, [rupino@dei.uc.pt](mailto:rupino@dei.uc.pt)

Antonio Dias de Figueiredo  
*Universidade de Coimbra*, [adf@dei.uc.pt](mailto:adf@dei.uc.pt)

Follow this and additional works at: <http://aisel.aisnet.org/amcis2000>

---

### Recommended Citation

da Cuhna, Paulo Rupino and de Figueiredo, Antonio Dias, "Information Systems Design Under a Different Light" (2000). *AMCIS 2000 Proceedings*. 257.  
<http://aisel.aisnet.org/amcis2000/257>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Information Systems Design Under a Different Light

Paulo Rupino da Cunha, DEI/CISUC, Universidade de Coimbra, rupino@dei.uc.pt  
António Dias de Figueiredo, DEI/CISUC, Universidade de Coimbra, adf@dei.uc.pt

## Abstract

This paper presents a new information systems design method conceived to accommodate emerging design trends and face the increasing complexity of information systems. It takes into account the growing business pressures that are put upon IS designers, the rising appeal of ready-made software (such as ERP and CRM systems), the growing weight of intranets and extranets, the need to account for legacy, and the call for effective management of the evolving portfolio of heterogeneous solutions that the "information systems" of the present day have become. It follows an approach that is quite different from those found in traditional information systems planning, in that it does without detailed information systems architectures and uses instead an identification of organizational entities whose responsibilities towards their environment may be supported by a variety of systems alternatives. The results thus produced enable a clear view of the present and future direction of the information system, while retaining a close relationship to the original business needs and the actual deployment options. A quick reference is made to a tool developed to support the method.

## Introduction

Information systems in organizations, once heavily centered on custom-coded software solutions, now increasingly rely on ready-made software products that satisfy most of the standard back-office and front-office needs. The code that remains to be written tends, at present, to concentrate mainly on supporting distinctive business facets, customizing packaged software, and performing integrations across the several components of the "whole" information system, which may include a significant amount of legacy (Asbrand 1999; Stephens 1998; Ward and Peppard 1996).

The increasing importance of intranets and extranets also contributes to change significantly the traditional concept of information system design.

This evolution of information systems seems to result from a series of emerging challenges. The pressure of competition imposes faster deployment times, as well as differentiated priorities and resource allocations, according to the business aspects that are to be supported. This suggests differentiated sourcing strategies for diverse systems needs. The current, highly sophisticated, readily-

available and diversified software market, with a wide range of offer, tends to render infeasible most of the traditional, long-coding, projects. The portfolio of heterogeneous systems and solutions, in which companies have invested over the years (the *legacy*), and on which they depend, calls for adequate management and leveraging mechanisms, instead of being ignored, as is the case with some information systems planning methods (Zachman 1993-1996).

Under these circumstances, the architectures of data and processes, or of objects, that information systems planning methods traditionally provide are not enough. Indeed, as most of their deliverables are strongly software-development oriented (Gale and Eldred 1996; Jacobson et al. 1995; Martin 1990; Taylor 1995), they tend to present a low level of granularity that impairs an holistic view.

The need to change the focus from the "delivery of technology" to the "delivery of benefits" by information systems projects, as pointed out by Ward (Ward and Peppard 1996), should be kept in mind.

A higher granularity view is, thus, needed to let us group existing systems together with new ones, acknowledge and combine the full range of sourcing alternatives (e.g.: package acquisition, outsourcing or development), and model more closely the interactions in the business that influence intranet and extranet design. This higher granularity should afford increased visibility and awareness of the "whole" information system.

Our method attempts to meet these demands, aiming at the design of the "whole" information system of the organization. Meant to be used in an ongoing manner, it delivers at any time a "roadmap" of the information system, with existing and planned subsystems, along with recommended sourcing strategies for each (e.g.: purchase, outsourcing, prototyping or custom coding), as well as relative priorities as dictated by business concerns. The relationship of the various subsystems to the organizational needs they support is also preserved.

As the approach is relatively light, the "roadmap" is fluid and can be reshaped in tandem with the way the organization itself evolves as a consequence of its internal strategy and environmental conditions.

We have followed Checkland's Soft Systems Methodology (Checkland and Scholes 1990), recognizing that this research approach is particularly suited for the

task (Avison et al. 1999; Baskerville 1999). The results presented here express the findings of our first iterations.

In the following sections, we start by delving into the reasons why an integrated approach, centered on higher granularity, is important. We then move on to the presentation of the proposed method, discussing the conceptual and field approaches and the deliverables, and briefly presenting a tool developed to assist its application. Finally, we draw some conclusions.

### The need for an integrated approach

When we observe a number of real world information systems environments, across radically different organizations (distinct businesses or even non-profit institutions), a common pattern becomes apparent: the "whole" information system is, in fact, made up of a number of autonomous components. We find some largely proven and reliable subsystems, often expensive packaged software solutions (customized to match the practices of the organization), some custom coded subsystems that cover issues that are very specific to the organization, and several "common" applications such as word processors, spreadsheets and other shrink-wrapped software.

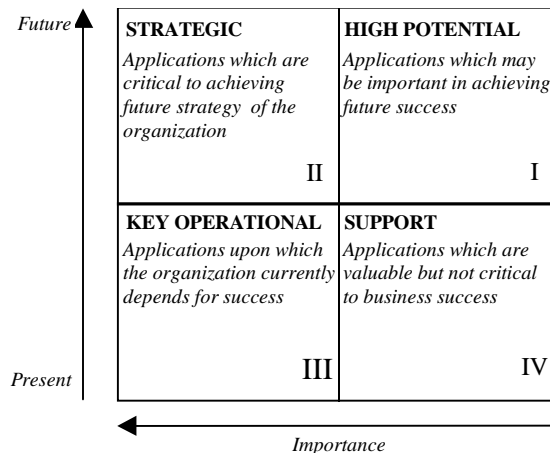
In this heterogeneous environment, several different vendors are generally involved. By using internal skills or through outsourcing, a certain degree of articulation among some subsystems is often put in place.

We also notice that new promising trends begin to materialize, such as Application Service Providers (ASP) that propose to host and manage the most critical applications (Booker 1999; Keegan 1999; Mateyaschuk 1999; Nickell 1999; Seymour 1999).

The deployment scenarios just described are in clear contrast with the results obtained from most traditional information systems planning methods, where blueprints of processes and data, or of objects, are much more oriented towards custom development than towards the articulation of distinct sourcing strategies for diverse information system components in agreement with their relative business importance (Gale and Eldred 1996; Jacobson et al. 1995; Martin 1990; Taylor 1995). This mismatch is generally felt when moving from plan to implementation, with quite problematic gaps arising between the results of planning and the solutions that can feasibly be deployed to the real world.

Our alternative approach departs from McFarlan's strategic grid (McFarlan 1984). The grid distinguishes four different categories of system, according to their contribution to the business, as shown in Figure 1.

Figure 1. McFarlan's strategic grid



The exploration of the original grid concept has been refined in (Edwards et al. 1995; Ward and Griffiths 1996) to enable detailed analysis of the application portfolio of a company. The classification that results from placing into the grid the various applications that make up the "whole" information system is used to help establishing the allocation of resources (such as people, technology, time and money) for each case. The most adequate strategies to obtain subsystems in each of the four categories corresponding to each quadrant are also presented in (Edwards et al. 1995), and reproduced in Table 1.

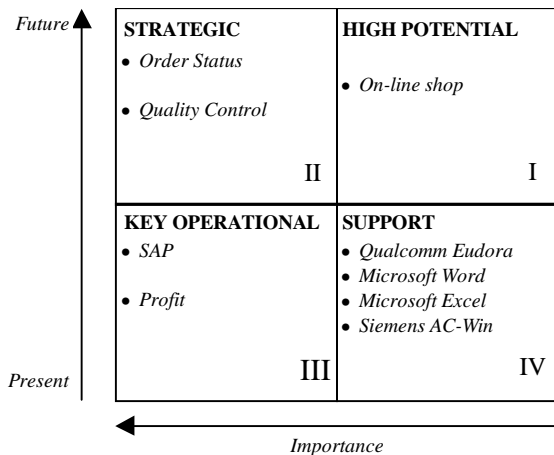
Table 1. Sourcing strategies according to the quadrant of the strategic grid

Grid Quadrant	Key issues	Sourcing Strategy
High Potential	Quickly assess potential benefits of the proposed system and decide on subsequent project destiny.	Rapid prototyping.
Strategic	The system importance is already clearly defined and related to business strategy. The main risk is missing a window of opportunity, so the ideal technological solution is constrained by the business objectives.	Rapid application development.
Key Operational	Proven technological solutions, resulting from years of experience, should be used, even if some organizational expediency sacrifices are needed	Increasingly satisfied using expensive packages, purchased using a terms of reference document, to ensure stable, long-life, effective solutions.
Support	Emphasis is on low cost, long-term solutions. Compromising user needs to the solutions available off-the-shelf is acceptable.	Purchase of shrink-wrapped software.

Equally important is the fact that information systems are not static artifacts. The position of their various components within the grid changes with time in response to changes in the variables that affect the business itself (Bhabuta and Veryard 1989; Edwards et al. 1995; Ward and Griffiths 1996). Those components frequently start at the *high-potential* quadrant, in support of promising business ideas, then progress to the *strategic* quadrant, if those ideas become sources of competitive advantage, and finally end up in the *key-operational* quadrant, when the competition catches up.

By applying the strategic grid (for illustrative purposes only) to a simplified information system, we may obtain a characterization as shown in Figure 2.

Figure 2. A typical information system characterized according to the strategic grid



In the *high potential* quadrant we find a subsystem aimed at selling the manufactured products over the Web - a business opportunity that the company wants to evaluate. In the *strategic* quadrant we find a subsystem that allows preferred customers to check the status of their orders over the internet (big orders take a considerable time to produce), and a second system that enables an improvement in the way the company presently performs the quality control of the products it manufactures. Both these initiatives are closely tied to the company's strategic objectives.

If the e-commerce venture of this company turns out to be successful, then the *On-line shop* subsystem will move to the *strategic* quadrant, as the initiative will become the source of competitive advantages. Resource allocation for the evolution and maintenance of the subsystem will be appropriately revised. As for the items already listed in the *strategic* quadrant, they will, in time, become *key operational*, as competition matches the added value that these systems currently represent.

In the *key operational* quadrant we find SAP, a popular enterprise resource planning (ERP) application, and another proven, reliable, system for factory automation. Finally, in the *support* quadrant, we find some standardization on shrink wrapped software as well as other "minor" items, such as an application that gives access to data from the private branch exchange (PBX), to control phone cost allocation and usage.

From the above example, four issues become apparent:

1. The mix of approaches actually used in the field, by information technology departments, to source different information system parts, is much closer to

the business-driven orientations described in McFarlan's strategic grid framework than the architectures that result from most traditional information systems planning methods, as these are frequently "biased" towards coding. Their focus is also frequently too narrow to adequately encompass all four systems categories.

2. Those same detailed architectures do not seem to fit well any of McFarlan's grid quadrants. According to Table 1, the most favorable cases seem to belong to quadrant I and quadrant II. Yet, in the first case, the very nature of the prototyping technique counters a previous phase of detailed specification; in the second case, while the detail produced in most architectures tends to seem excessive to the non-specialist elements of the design team, it is rarely sufficient to avoid the use of more accurate software engineering techniques at a later stage. Besides, in many cases, a prototype originating from the *high potential* quadrant is already available as a basis for *strategic* systems, providing a much richer specification than those provided by most information system "architectures".
3. The dynamic nature of the various subsystems, which causes the movement across the grid quadrants during their life time (and a consequent impact on resource allocation) is frequently "forgotten" by information system design methods, that give no clues on how to handle the life cycle. They usually fail to indicate clearly which of the four types of subsystems they were designed to model. They often address only some type, hindering the visibility across the "whole" information system, and they can even get to the extreme of not recognizing any difference between information system components, thus attempting to apply equally to the subsystems in all quadrants of the grid, regardless of the different allocation of resources that they require. This reality is, to some extent, a consequence of the excessive influence of technically minded software engineering techniques over business concerns, that still prevail in most information systems planning methods.
4. "Legacy" systems, that tend to be systematically disregarded or minimized in traditional information systems planning, often represent the biggest share of the "whole" information system, as the population of the grid shows. This suggests that their leveraging and integration should be carefully taken into account. Actually, the "legacy" should be viewed as a regular and valuable part of the information system, as a whole, and accounted for in the overall evolution plans. Failure to do so will cause an increase in "legacy" at every new project (Zachman 1993-1996).

## An holistic design method

To face the challenges that we have pointed out, information systems design methods should adopt an *integrated view* and balance technical and business considerations more evenly. The business pressures caused by increased competition, and their impact on the evolution of the information system, should be acknowledged. The past investments – represented by legacy – should be considered on equal footing with the new systems. Also, distinct resource allocation strategies should be considered from the onset, so that package acquisition and outsourcing (or even newer application service provision - ASP) can be regarded as natural alternatives for systems sourcing.

As the participation of non-specialists in the design of information systems is becoming increasingly important, the methods should be as understandable as possible by those people, as well as sufficiently light to let design results evolve in tandem with the organization, thus reflecting accurately the "roadmap" of the information system.

The design results should not only avoid leaving gaps to actual field deployment, but actually go one step further, acknowledging and assisting in the choice between available alternatives.

### *The conceptual approach*

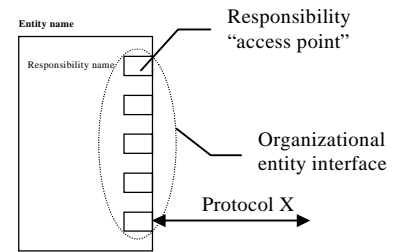
To meet these challenges, we propose to handle organizations and systems design from a new viewpoint: *at the level of the organizational entity*. For each organizational entity, we propose the identification of the *responsibilities* it holds towards its environment and of the *protocol* that must be used for *interaction with the organizational entity through those responsibilities*.

An *organizational entity* can represent several organizational realities, ranging from the clearly defined "divisions", such as "human resources", typical of more mechanist organizations, to versatile divisions or even teams, characteristic of more organic types of organization (Morgan 1997).

Returning to the "human resources division" to illustrate to concept of organizational entity, we see, for example, that it must provide a *service* for workers to justify their absences, perhaps by filling an internal form and enclosing a medical justification. All such "major" *services* that can be asked of an organizational entity are the responsibilities to be modeled.

The representation of an organizational entity is depicted in Figure 3.

Figure 3. Representation of an organizational entity



Rather than attempting to describe rigorously the internal processes of each organizational entity, our proposal deliberately avoids them, focusing instead on the precise definition of all the "access points" through which interaction with the entity is possible. This responsibility orientation is well supported theoretically and has been successfully applied in different contexts (Wirfs-Brock and Wilkerson 1989; Wirfs-Brock et al. 1990).

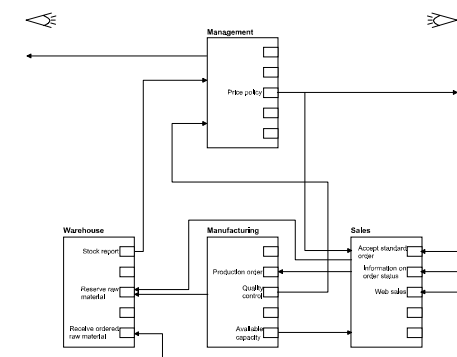
Two illustrations may help clarify our approach:

- To perform their everyday duties, workers do not really need to know, or care, about how the various organizational entities carry out their procedures internally. What they do need to know is how responsibilities are distributed inside the organization and what protocols must be activated when they need to perform their own tasks.
- Moreover, many of the responsibilities of the organizational entities can often be satisfied by ready-made applications. Actually, the detail offered by the architectures that result from many traditional information systems planning methods is frequently excessive and useless, as some of the logic or data structure they express will already be embedded in a software solution that is likely to be purchased or outsourced.

Recalling again the above example of the human resources division, we should note that elaborate descriptions of the internal procedures in terms of objects or data structures and processes may prove excessive, as it is very likely that the data supplied by the worker (at the responsibility interface) will end up being entered to a "standard" human resource module of an enterprise resource planning (ERP) package.

By collecting the right information about the various organizational entities, it is possible to establish how they use each other's services (responsibilities) in order to fulfill the broader business objectives, as illustrated in the simplified example diagram of Figure 4.

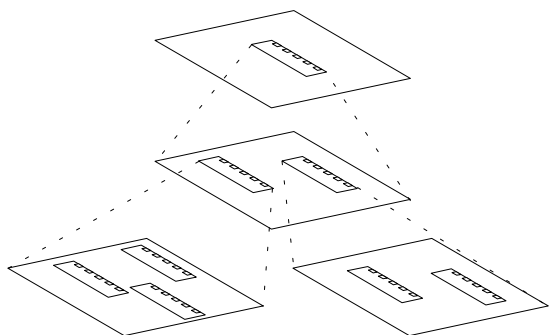
Figure 4. The business modeled as a set of organizational entities interacting to fulfill the broader business objectives



It should be noted that the organizational entity responsibilities we describe *do not necessarily have to benefit from information system support* at the present time. All responsibilities towards the environment of the organizational entities are modeled regardless of this aspect. The outcome of this procedure is, in fact, an *organizational model* where information system support is (partially) *blended* in. It also becomes closely tied to the business need it supports.

Of course, the complexity of most organizations is far greater than can be modeled in a planar diagram like that of Figure 4. To cope with this, the proposed method uses several abstraction layers. First, the entire organization is modeled as an organizational entity that interacts with the external environment; then, it is "exploded" into a next level of organizational entities where the upper-level responsibilities acknowledged by the environment are delegated to the concrete lower level entities that actually handle them. This "leveling" process continues for each organizational entity until manageable levels of complexity are reached. This is illustrated in Figure 5.

Figure 5. Recursive decomposition of organizational entities into their components



The identification of the entities in each step of leveling is straightforward, since the decomposition follows the existing organization.

The decomposition into several levels of abstraction also serves to address the human cognitive limitations that cause our performance to degrade when we have to keep in mind more than 7 to 9 elements at the same time (Miller 1956). This concern is also present in proposals found in (Taylor 1995) and (Gale and Eldred 1996), but is missing in other methods, such as Information Engineering (Martin 1990) and Business Systems Planning and its derivatives (IBM 1984; Spewak and Hill 1992).

### The field approach

Although it is commonly acknowledged that any procedures and documents involved in the design process must be accessible to non-specialists, namely the managers whose understanding and endorsement is critically necessary for any large-scale project, people inside organizations still complain that the practices and materials generally used remain far too complex (Davenport 1997).

To overcome this difficulty, we have chosen CRC (class responsibility collaborator) cards (Beck and Cunningham 1989) as our field instrument. CRC cards are well known for their pedagogic and conversational qualities, that allow non-specialists to quickly contribute to complex projects and produce valuable results (Cunningham 1994; Mitchell 1997; Taylor 1995; Wilkinson 1995). A natural characteristic of the method we propose, that of centering the dialogue on concepts and issues that relate to everyday work, is amplified by the use of this technique, in the sense that it narrows the communications gap and consequently lets people adhere more easily to the planning projects and provide significant contributions earlier. Figure 6 shows the "front" of a CRC card used in our method. A card corresponds to an individual organizational entity.

Figure 6. A simplified CRC card for use in business modeling

Organizational Entity Name:	Upper level Organizational Entity:
Responsibilities	Clients
Composing Organizational Entities	

To accommodate the modeling of the organization we have introduced changes to the original cards: instead of identifying, for each responsibility, whose collaboration it needs, we take a more customer-centered approach and ask, for each one, "whom does it serve". This is consistent with the desired encapsulation of the way responsibilities are handled inside organizational entities. Some additional

information is also recorded on the "back" of the card, such as:

- Details regarding the protocol used to access each responsibility;
- A description of the organizational aims and procedure underlying each responsibility;
- A description of any technological infrastructures already in place to support each responsibility (the legacy);

It is also at this stage that the requisites of future information systems support are stated. Those new or revised requirements may lead to brand new systems for previously unsupported responsibilities or to the improvement of existing systems. Finally, a key issue when describing each responsibility is to classify, in terms of the strategic grid, the significance of the information system support it requires, which is done according to the importance the responsibility holds towards the business goals. This information is used later to obtain the correct choice of priorities and sourcing strategies for the various information system components.

### The deliverables

The main results from the application of the method take the form of diagrams like those of Figure 4 and Figure 5, that model the organization and support consensus building about interactions, CRC cards that facilitate continuous dialog, and text reports that consolidate, recombine and group the various pieces of information that have been collected for every responsibility. The descriptions of the business procedures underlying the responsibilities, together with those regarding (existing and required) technological aspects provide the foundation for the various approaches to deployment. The choice between approaches is carried out using the *significance* that has been established in terms of the strategic grid, as collected for each responsibility. Depending on this parameter, the information collected for each responsibility may be used differently: as the *terms of reference* for the acquisition of *key-operational* systems; as a statement of requirements for the team that takes over and proceeds with adequate software engineering techniques – such as *fast prototyping*, in the case of *high-potential* applications, or *rapid application development*, in the case of *strategic components* – and, finally, as the guide to a low-overhead market survey for subsystems belonging to the *support* quadrant.

The grouping of the requirements established across the individual organizational entities, according to the four categories of information systems considered in the grid, lets the responsibilities held by the different entities (such as, e.g., warehousing and accounting) to end up being satisfied, if necessary, by the same information

systems solution, such as an enterprise resource planning (ERP) package that covers multiple areas.

A particularly useful kind of report is obtained by viewing the "information system" mapped to a McFarlan grid (as in Figure 2), providing visual insight about the present and planned subsystems.

In addition to the recombination of the collected information into various types of reports, additional data can be extracted from the model. Indeed, for each organizational entity there are always two viewpoints, as represented by the "eyes" in Figure 4.

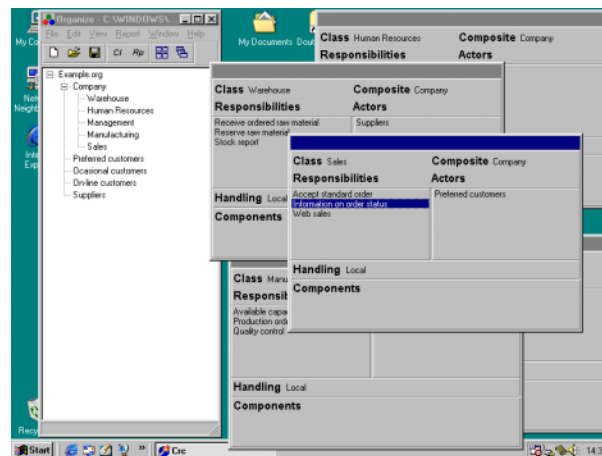
The viewpoint on the right reflects the complete set of services from other organizational entities that the entity under analysis needs to access in order to fulfill its own responsibilities. This information can be used to specify exact access profiles to whatever solution is adopted. Custom user interfaces can also be derived that give access only to the responsibilities from others that each organizational entity effectively needs.

The viewpoint on the left expresses the services offered by a given organizational entity and the corresponding *load* upon the entity, as conveyed by the number of links that diverge from it.

### A tool to assist the method

A user friendly software tool has been developed to assist in the exploitation of the method. It is based on customized CRC cards and supported by a relational data base that stores the collected data. This enables diversified queries to lead to reports that provide different views of the information extracted from the model. Space constraints inhibits us from presenting a complete application example. Nevertheless, the following figures provide a simplified overview that illustrates how the application and the method work together.

Figure 7. General view of the modeling tool

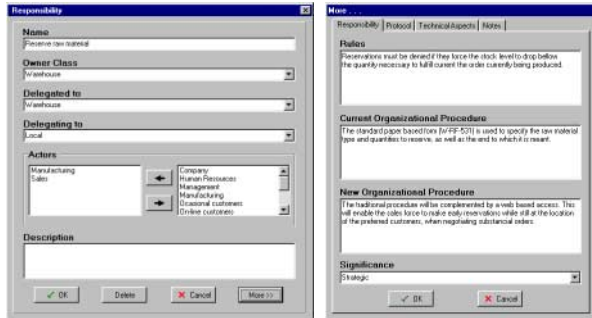


On the left of Figure 7 we observe a tree view showing the relationship between organizational entities and on the

right side several CRC cards are shown, one for each organizational entity listed on the tree. By selecting any responsibility in a card, it is possible to see the list of the clients for that service. From this information, the links between entities can be derived.

By using the right mouse button over any responsibility we have access to the "back" of the card (Figure 8), where details are recorded. Various "tabs" let us have access to the various categories of information discussed at the end of section regarding the field approach.

Figure 8. Property dialogs for the responsibilities



## Conclusion

Information systems planning in organizations is facing new challenges to which traditional methods fail to respond. To meet these challenges, we have proposed a method that radically departs from the traditional practices, where detailed architectures of data and processes (or objects) are produced. Our approach *brings together* organization and information systems modeling around *responsibilities* and *organizational entities* that are accountable for them. The internal functioning of each organizational entity is deliberately ignored, while every "access point" through which the service of an organizational entity can be requested is rigorously and exhaustively defined. The organizational entities can only interact with each other through these interface points. The growing use of intranets and of ready-made software packages that cover a wide range of responsibilities tends to render useless the finer detail of traditional methods, thus endorsing this approach. The integration of the legacy "inside" organizational entities also becomes much more natural and seamless.

Additional advantages emerge from the closeness between the model and organizational reality: the questions posed to the members of the organizations during the design phases are centered on concepts and issues that are familiar to their everyday routine. This effectively narrows the communications gap, when compared to other methods that introduce "artificial" modeling elements or excessively technical ones.

The method acknowledges distinct contributions to the business made by distinct information system components, thus facilitating a clear definition of priorities for deployment. This embedded classification mechanism enables the *integrated* modeling of the *whole* information system, with obvious advantages in terms of increased awareness, without the risk of being overwhelmed or distracted by the great number of elements that are identified.

A software tool has been developed to assist in the exploitation of the method. A second version of this tool is now underway, to reflect lessons learned from the field.

At this stage, one full cycle of research, as described in the methodology we have been following – the Soft Systems Methodology – has been completed. The first field case situation consisted of the evolution of an existing information system for a public institution. A document has been produced that provides a clear view over the "whole" information system and prescribes a series of interventions in various organizational entities of the institution. A particularly important aspect has been the design of an intranet that integrates and leverages dispersed autonomous subsystems into a coherent whole. This particular issue has been much facilitated by the strong emphasis that the method places on interactions occurring in the environment. The viewpoints described in the deliverables section have been used to derive customized user interfaces to the intranet, favoring the access to other services that each organizational entity really needs.

To comply with Soft Systems Methodology, further iterations of the research process should now follow, to test adjustments and evolutions to the proposed method. Three new real world cases are already scheduled to this end.

## Acknowledgements

The work described in this paper has been partially supported by the Portuguese Foundation for Science and Technology (FCT) under research contract 326/94 and by PRODEP under grant 5.2-nº1/96.

## References

- Asbrand, D. "Legacy systems: The keepers and the goners," ERP Hub - Interface Development, [www.erphub.com/interface\\_p1.html](http://www.erphub.com/interface_p1.html), (Current May 1999).
- Avison, D., Lau, F., Myers, M., and Nielsen, P. A. "Action Research," *Communications of the ACM*, (42:1), 1999, pp. 94-97.
- Baskerville, R. L. "Investigating Information Systems with Action Research," *Communications of the Association for Information Systems*, (2:19), 1999.



- Beck, K., and Cunningham, W. "A Laboratory For Teaching Object-Oriented Thinking," *OOPSLA'89*, c2.com/doc/oopsla89/paper.html, (Current Apr. 23, 1998).
- Bhabuta, L., and Veryard, R. "Relating IT/IS Planning and Strategic Management to Organizational Effectiveness," Proc. International Conference on Organization and Information Systems, Z. Kaltnekar and J. Gricar, eds., Bled, 1989.
- Booker, E. "ASP Model Gains Legitimacy," Internet Week, www.internetwk.com/story/INW19991028S0012, (Current Oct. 28, 1999).
- Checkland, P., and Scholes, J., *Soft Systems Methodology in Action*, Wiley, Chichester, 1990.
- Cunningham, W. "How Do Teams Shape Objects? How Do Objects Shape Teams?" *OOPSLA'94*, c2.com/doc/oopsla94.html, (Current Apr. 23, 1998).
- Davenport, T. H. *Information Ecology: Mastering the Information and Knowledge Environment*, Oxford University Press, New York, 1997.
- Edwards, C., Ward, J., and Bytheway, A. *The Essence of Information Systems (2nd Ed.)*, Prentice-Hall, London, 1995.
- Gale, T., and Eldred, J. *Getting Results with the Object-Oriented Enterprise Model*, Sigs Books, New York, 1996.
- IBM. "Business Systems Planning," *GE20-0527-4*, IBM, 1984.
- Jacobson, I., Ericsson, M., and Jacobson, A. *The Object Advantage: Business Process Reengineering with Object Technology*, Addison-Wesley, Wokingham, 1995.
- Keegan, P. "The Death of Software?," Upside Today, www.upside.com/texis/mvm/story?id=37d936dc0, (Current Sept. 21, 1999).
- Martin, J. *Information Engineering, Book II: Planning and Analysis*, Prentice-Hall, Englewood Cliffs, 1990.
- Mateyaschuk, J. "Leave The Apps To Us! -- ASPs Offer Benefits Through Economies Of Scale," InformationWeek, www.techweb.com/directlink.cgi?IWK19991011S0032, (Current Nov. 12, 1999).
- McFarlan, F. W. "Information Technology Changes the way you Compete," Harvard Business Review, May-June, 1984, pp. 98-103.
- Miller, G. A. "The Magic Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information," *Psychological Review* (63:2), 1956, pp. 81-96.
- Mitchell, I. (1997). "The Card Deal," OMO - Object Magazine Online, www.sigs.com/omo/tips/9705/9705.mitchell.html, (Current Apr. 24, 1998).
- Morgan, G. *Images of Organization*, SAGE Publications, Thousand Oaks, 1997.
- Nickell, J. A. "ASPs: Software Superstores of the Future?," webreview.com, webreview.com/pub/1999/09/03/feature/index2.html, (Current Sept. 3, 1999).
- Seymour, J. "Send Out for Software," PC Magazine, www.zdnet.com/pcmag/stories/reviews/0,6755,2344646,0.html, (Current Oct. 13, 1999).
- Spewak, S. H., and Hill, S. C. *Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology*, Wiley-QED, New York, 1992.
- Stephens, E. "Ready, Set, Go! (Again)," Software Development, 1998, pp. 38-44.
- Taylor, D. A. *Business Engineering with Object Technology*, John Wiley & Sons, Inc., New York, 1995.
- Ward, J., and Griffiths, P. *Strategic Planning for Information Systems*, John Wiley & Sons, Chichester, 1996.
- Ward, J., and Peppard, J. "Reconciling the IT/Business relationship: a troubled marriage in need of guidance," *Journal of Strategic Information Systems*, (5:1), 1996, pp. 37-65.
- Wilkinson, N. M. *Using CRC Cards - An Informal Approach to Object-Oriented Development*, Cambridge University Press, SIGS Books, New York, 1995.
- Wirfs-Brock, R., and Wilkerson, B. "Object-Oriented Design: A Responsibility-Driven Approach," *OOPSLA'89*, 71-75.
- Wirfs-Brock, R., Wilkerson, B., and Wiener, L. *Designing Object-Oriented Software*, Prentice Hall, Englewood Cliffs, NJ, 1990.
- Zachman, J. A. "Enterprise Architecture and Legacy Systems - Getting Beyond the "Legacy"," www.ozemail.com.au/~visible/papers/zachman1.htm, (Current Jan. 19, 1998).