The Use of the Key Driver Technique in the Design of Copiers

W.P.M.H. Heemels[a], L.J. Somers[b], P. van den Bosch[c], Z. Yuan[c],
B. van der Wijst[d], A. van den Brand[e], G. Muller[a]

[a]*Embedded Systems Institute, PO Box 513, 5600 MB Eindhoven, The Netherlands*
maurice.heemels@esi.nl

[b]*Eindhoven University of Technology, Dept. Math. & Comp.Sc., Eindhoven, The Netherlands*

[c]*Océ Technologies BV, Venlo, The Netherlands*

[d]*Philips Applied Technologies, Eindhoven, The Netherlands*

[e]*Centric TSolve BV, Son, The Netherlands*

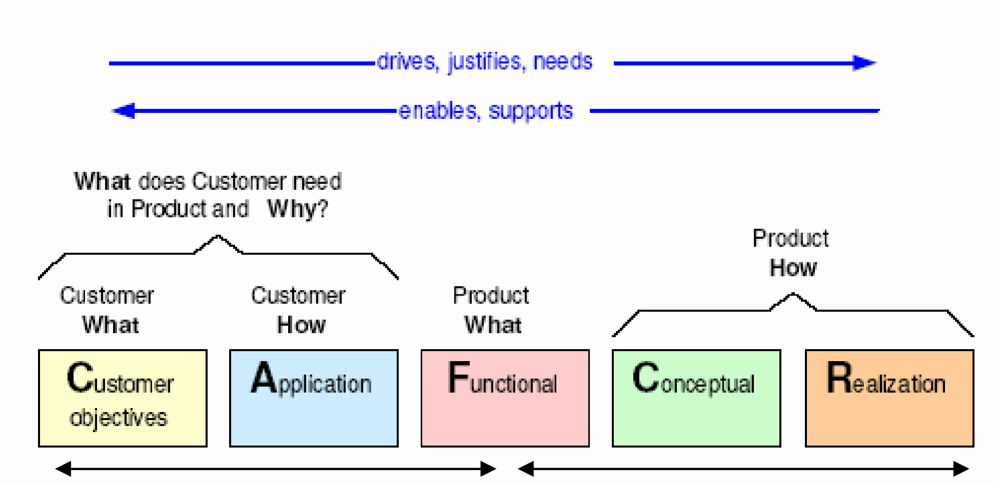
Abstract

In the design of many of the complex products we encounter today it is important to remain focused on the essential customer objectives, which can be captured in terms of customer key drivers. In order to obtain a successful product these key drivers need to be related to the system requirements. A structured overview of the relationships between key drivers and system requirements is an effective means to support the design. This paper proposes the key driver technique and its use to create such an overview that is also highly convenient in requirements tracing. A stepwise approach is presented together with guidelines on how to obtain a key driver model. An industrial case study for a high-volume copier demonstrates the application and serves as a proof-of-concept for the key driver technique. The benefits, limitations and lessons learned are presented.

1. Introduction

The complexity of the products being designed by industry today is increasing at an astonishing rate. To keep the design of complex machines focused, it is important to know the essential customer objectives and to relate these to those system requirements that have the largest influence on them. A structured graph showing these relations helps to keep an overview of the overall design. In particular, specifications can be easily traced back to the objectives of the customer. The *key driver technique* as explained below is one of the possible means to obtain the system requirements in a systematic manner and to provide such a structured overview.

The key driver technique fits in the broader framework of the "CAFCR" methodology [1], which is a decomposition of a system architectural description into five views, as shown in Figure 1. The *customer objectives* view (what does the customer want to achieve) and the *application* view (how does the customer realize his goals) capture the needs of the customer. The needs of the customer (what and how) provide the justification ("why") for the specification and the design of the product. The *functional* view describes the "what" of the product, which includes (despite its name) next to the functional also the non-functional requirements. The "how" of the product (the technical solution) is described in the *conceptual* and *realization* views. In this way CAFCR is focused on the relation between the customer world and the product. The functional view can be seen as the interface between the problem and solution world. Another dimension that plays a role in specification and design is the *manufactural* view. The manufactural view describes operational aspects of the product manufacturer (*not* the operational aspects of the customer that belong to the A view) like preferred way of designing and producing, culture, type and number of employees, etc. The job of the system architect is to integrate these views in a consistent and balanced way, in order to get a valuable, usable and feasible product.



"Problem" "Solution"

Figure 1. CAFCR views

The key driver technique, as will be explained in section 2, can be considered a "CAF submethod" of CAFCR. The key drivers represent the main customer objectives (C view). The key driver technique helps to derive the more detailed and quantified system requirements (F view). It translates a few (three to six) customer key drivers into maybe hundreds of system requirements. For instance, in the copier case study described in this paper, 3 key drivers will be expanded to like 40 system requirements. A bridge between key drivers and requirements is a layer called (derived) application drivers (typically representing the A view). The key and application drivers in the customer views (CA) will be linked via requirements (F) to design choices in the other views (CR). Structuring this information graphically helps to keep a good overview over the design process. It is useful for the designers to see why a certain requirement is important from a customer perspective or in other words, to understand the "why" behind the requirements / specifications they have to realize (traceability). The final system requirements are determined on one hand by the customer side (the objectives coming from CA views), but on the other hand just as much by the technical (im)possibilities and solutions side (CR) and the manufactural view. The key driver technique focuses on the CAF views, but, as the final key driver model is a living entity, the influence of the CR views grows during the design process.

Closely related to the key driver technique is Goal Oriented Design [2],[3],[4], which is also based on analysis of the customer needs and goals in a hierarchical fashion. A goal oriented approach like KAOS [3] or GRL [5] starts with goals and derives requirements from them. URN [6] combines use case maps [7],[8],[9] (manufactural and functional requirements) and GRL (non-functional requirements). Also in Quality Function Deployment [10], where the term "benefit" is used for key driver, the link between benefits, engineering requirements and design concepts is emphasized.

The key driver technique gains its value from relating a few sharply articulated key drivers to a much longer list of requirements. By capturing these relations, a much better understanding of customer and product requirements is achieved. The "why" behind requirements is documented and the focus is maintained on the most important issues and the trivial ones are left out. Other important values are its conceptual simplicity and the fact that all product creation phases and corresponding views are taken into account.

This paper describes the key driver technique and provides guidelines how to obtain a key driver model. Its application and effectiveness are shown for a high-volume copier. The goal of the industrial case study is to understand the end user requirements by building a key driver model. Moreover, the key driver technique is extended by a matrix that links the functional decomposition of the copier to the requirements obtained via the key driver technique. This extension offers the possibility to have an overview of the "responsible" functions for realizing a specific requirement as the matrix provides a first breakdown of a requirement into the functional subsystems. This gives a means to monitor the progress or can even be a starting point for further design support techniques, like budget-based design [10].

The outline of the paper is as follows. In section 2 the key driver technique is presented. In section 3 the case study of the copier is explained and the key driver technique is applied. In section 4 the final graphical overview of the key driver model is shown and discussed. In section 5 we combine the results of the key driver model with a breakdown of all the requirements with respect to the functional decomposition of the copier. Section 6 gives the strengths and limitation of the model and the lessons learned. We end with the conclusions in section 7.

2. Key driver technique

The key driver technique couples the customer side captured in the key drivers (Customer Objectives view) to the product side (Conceptual and Realization view) via the application driver (Application view) and system requirements (Functional view). The following ingredients play an important role:

- *Key drivers (C view)*: the top three to six driving customer objectives (What does the customer want?) For instance, a copier in a copy shop needs to print large volumes in a short time. Hence, productivity/volume is a key driver.
- *Application drivers (A view)*: Drivers that describe how the customer is realizing the key drivers (How does the customer achieve his goals?) Application drivers are closer to the solution space than the key drivers. For instance, the productivity/volume of a copier can be achieved by a.o. speed of printing and reliability. Speed and reliability are typical derived application drivers for productivity.
- *System requirements (F view):* Detailed (often quantitative) specifications of the product or its subsystems. For instance, the speed of printing can be related to the system requirement of 85 pages per minute (for A4 paper) in full production.

The C(ustomer objectives), A(pplication) and F(unctional) views are explicitly included in the key driver model. However, it does not *explicitly* include the C(onceptual) and the R(ealization) views (solution side) although these views have a strong influence on the final system requirements as well. Also the operational constraints and preferences of the manufacturer and other stakeholders than the main customers determine the final requirements. This information is *implicitly* included via the process of setting up the key driver model as described below e.g. via interviews with engineers or system architects and using core domain knowledge from the manufacturer's previous projects.

To obtain a structured overview of key drivers, applications drivers and system requirements, Figure 2 gives an outline of the key driver technique.

The *first step* is to define the scope of the key driver graph. From which customer or other stakeholders do we want to understand the objectives or needs? For the choice of the customer it is important to determine the market segment of the product. Also the system boundary plays an important role. For instance, for a copier, do we want to consider a stand-alone copier (with for example an office user as user) or do we consider a complete copy shop (with multiple networked copiers) for which the main operator or the director of the shop might be the main stakeholder?

The *second step* is to acquire facts, for example by extracting functionality and performance figures out of the product specification (for the predecessors of the to-be-developed system). Analysis of this information recovers implicit and hidden facts. The requirements of an existing system can be analyzed by asking "why questions" and mapping this to the new product. For example: "Why does the copier need additional turning of sheets?" At this point one might have an unstructured collection of various key and application drivers together with requirements.

The *third step* is to bring more structure in the facts, by building a graph, which connects requirements to key drivers. A workshop with brainstorms and discussions is an effective way to obtain such a graph. In this case, it is important to get the right people around the table representing the different views: marketing, strategic planning, system architecting and involved engineering disciplines. Also, interviews with people from previous projects or the current development project can be very beneficial in this stage.

• Define the scope specific.	in terms of stakeholder or market segments
• Acquire and analyze facts	extract facts from the product specification and ask why questions about the specification of existing products .
• Build a graph of relations between drivers and requirements by means of brainstorms and discussions	where requirements may have multiple drivers
• Obtain feedback	discuss with customers , observe their reactions
• Iterate many times	increased understanding often triggers the move of issues from driver to requirement or vice versa and rephrasing

Figure 2. Technique to link key drivers to requirements by iterating over four steps

The *fourth step* is to obtain feedback from customers. The total graph can have a lot of many-to-many relations, i.e. requirements that serve many drivers and drivers that are supported by many requirements. The graph is good if it is as simple as possible and the customers are enthusiastic about the key drivers and the derived application drivers. If a lot of explanation is required, then the understanding of the customer is far from good.

Frequent iterations over these steps improve the quality of the understanding of the customer's viewpoint. Each iteration causes some movements of elements in the graph in driver or requirement direction and also causes rephrasing of elements in the graph.

The use of the key driver technique benefits from the following guidelines:

- The most important goals of the customer are obtained by limiting the number of key drivers. In this way the participants in the discussion are forced to make choices.
- The focus in product innovation is often on differentiating features, or unique selling points. As a consequence, the core functionality from the customer's point of view may get insufficient attention. For instance, consider cell phones that are overloaded with features, but that have a poor user interface for making calls. The core functionality must be dominantly present in the graph.
- The naming used in the graph must fit in the customer world and should be as specific as possible. Very generic names tend to be always true, but they do not help to really understand the customer's viewpoint.
- The boundary between the customer objectives view and the application view is not very sharp. When creating the graph that relates key drivers to requirements, one frequently experiences that a key driver is stated in terms of a (partial) solution. If this happens, either the key driver has to be split, rephrased, or the solution should be moved to the requirement (or even realization) side of the graph. A repetition of such iterations increases the insight in the needs of the customer in relation to the characteristics of the product. Why, what and how questions can help to rephrase drivers and requirements.

In the next section, the key driver technique is illustrated by applying it to an existing copier.

3. Case study of a high-volume copier

This application context is characterized by document printing systems that are productive, reliable, and user-friendly. These systems can print on papers of several formats with different weights, sizes and fabric, automatically on both sides and including stapling, booklet production, or other types of finishing. In order to be perceived as reliable devices, such copiers must be very robust with respect to variations in media. Moreover, variations in timing that relate to paper transport must be controlled up to a high degree. As the printing speed is rather high (typically above one image per second), timing requirements are tight and advanced mechatronics are indispensable. This becomes even more apparent if one realizes that the positioning of images on the sheets has tolerances below one mm. Figure 3 presents an overview of a copier together with a decomposition into its major subsystems:

- *Scanner* module (SCAN): scans hardcopy sheets and produces digital images out of it.
- *Image processing and job control* (CONTROL): generation / adaptation of the digital images coming from the scanner (copy) or network (print) and scheduling of the print jobs (e.g. order of printing).
- *Paper input module* (PIM): trays from which sheets are separated and sent into the registration module.
- *Registration module* (REG): paper path that transports and performs accurate positioning of the sheets.
- *Print engine* (PRINT): receives the digital information that it transforms into a toner image and fused on the sheets.
- *Finisher (*FIN): collects all finished sheets.
- *User interface* (UI): the communication means between copier and user.

3.1 Step one: define the scope specific

For step one it is necessary to find out for which (group of) customers the key drivers are considered. Therefore, before thinking of key drivers and requirements, we first focus on the main stakeholders of the copier.

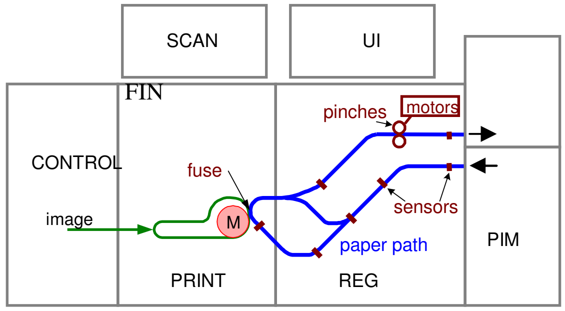


Figure 3. Schematic overview of a copier

3.1.1. Stakeholders

The high-volume copier under study was aimed at the market segment of the copy shop or the central document production (CDP). The main stakeholders are summarized in Figure 4. Both the copy shop and the CDP are characterized by a unit (an independent shop or a central place within a company) where individual customers or employees can go to get copies of their originals. Originals vary from simple sheets to entire workbooks.

One can distinguish the following customer type stakeholders for the CDP (see figure 4):

- Customer: the customer that goes to a CDP to get some copies or prints.
- Operator: this is a professional who uses the system for professional and productive (re)production of end documents. He also plans the work and prepares the jobs.
- Assistant Operator: a professional who uses the system for the professional and productive (re)production of documents. Jobs are processed and planned by the operator.
- System Administrator: an IT-professional responsible to keep the system networked.
- Buyer: responsible for acquiring new equipment.

Next to these user environment stakeholders, one also has the stakeholders for more conceptual, realization and manufactural views.

- Service department: the engineers that service the copier. They come to maintain the copier on a regular basis or in case of malfunctioning.
- Government: this stakeholder states restrictions concerning e.g. the environment (pollution, noise, energy usage, etc.) and safety.
- Development: the people that create the copier.
- Company board: board of the copier's manufacturer that is interested in the business success of the copier.

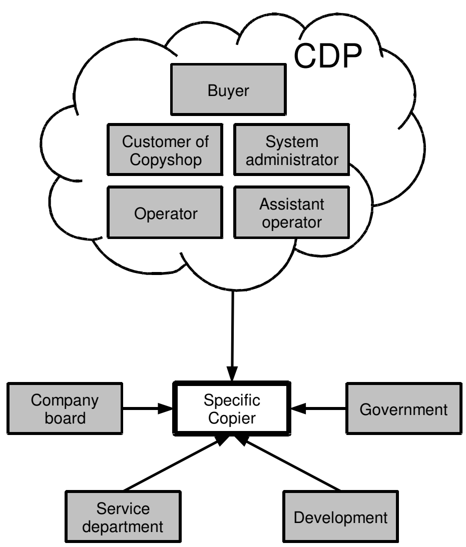


Figure 4. Overview of stakeholders

3.1.2 Selection of the stakeholders

Considering all the stakeholders, some focus is needed. As such, we concentrate on the stakeholders of the copier/copier depicted in the cloud in the top of Figure 4, as they represent the customer side.

3.2 Step two: Acquire and analyze facts

For each of the stakeholders, scenarios have been generated. Scenarios extend use cases (see e.g. [2],[7],[8],[9]) that are well known in software engineering. In the context of real-time software systems, scenario-based requirements analysis is also used in [12]. Scenarios for complete systems including hardware and mechatronics are very useful to find the essential customer objectives and needs (both qualitatively and quantitatively). The copier manufacturer has a set of such scenarios specified for each user, which turned out to be very helpful.

3.2.1. Scenarios for specific stakeholders

Due to space limitations we cannot present the scenarios for each stakeholder. Instead we concentrate the **operator** of the CDP. Typically from his perspective, document appearance and finishing quality (reproduction quality, stapling, etc) are equally important.

- He is the 'white collar' CDP worker (as opposed to the 'blue collar' assistant operator).
- He is an experienced and eager user of up-to-date repro IT-systems and workflow management tools.
- He prepares, plans, and produces high quality documents, with much variety in the type of documents. His main goal is to fulfill the needs of the CDP customer as good as possible. In this respect he is a service provider.
- He uses the system both at the control panel and through specific server workstations.
- His main job is working with the copier as productive and efficient as possible.
- He knows everything about the copier and has been certified by the copier manufacturer. Might be consulted by the copier manufacturer to define new products.

In the first exploration to get to the key driver model, we used scenario-based reasoning using the information above for the operator and we studied the available (public) commercial info and technical information. We combined this with a brainstorm session with people from the copier manufacturer, which gave rise to the first guesses on the key drivers. The initial guess of the key drivers for the CDP, based on expertise and previous projects, was:

- Productivity.
- Print quality.
- Integral cost per copy.

3.3 Step three: Build a graph of relations

During the brainstorm session we tried to map the identified key drivers onto the derived application drivers, which in turn can be translated into customer requirements of the copier. This is going from left to right in the CAFCR model in Figure 1. Because we experienced that determining the derived application drivers can be hard, we also tried to get them by first making an inventory of the technical system requirements and then translating them back to one or more derived application drivers. This is going from right to left in the CAFCR model.

To refine the initial key driver model, we interviewed the project leader of the development project for this specific copier. The project leader pointed also towards "ease-of-use" as one of the key drivers. This did not seem to be one of the *key* drivers for the CDP as there are professional operators in the CDP knowing the machine in all its aspects. As the project leader insisted on the extreme relevance of ease-of-use, we continued the discussion to which customer this was relevant (as we found it inconsistent with the CDP). The discussion led to office users as important end users of the copier, revealing that the copier was aimed at more than one market segments: next to the professional CDP, the copier is also intended as walk-up office copier as found in the aisles of many offices, where people make their own (limited number and limited complexity) copies or print-outs. Hence, the copier seems to have a "double personality": *one* product is aimed at *two* segments. Although this aspect required further attention, this interview confirmed the initial key driver model for the CDP part.

To get more insight in the "double personality" of the copier and to refine the model, we interviewed a person from the business strategy department. Time to market and cost of development were also very important (opportunity and market share). We decided not take these into the key driver model, as they are not directly related to the customer world. These issues are related more to the operational issues of the manufacturer. The business strategy perspective showed (as to be expected) the major differences in priority of the drivers and requirements as compared to the project leader or the end user. Important for us was that the interview with the business unit confirmed the two different market segments (CDP and walk-up) for one and the same copier. As a consequence, it was necessary to split the customers in two groups and include them both explicitly in the key driver model. This required a reconsideration of step 2.

3.4 Step two: Acquire and analyze facts

As mentioned in the previous section, it was concluded that indeed an additional branch of customer representatives (related to walk-up) was needed. To include the corresponding additional stakeholders, Figure 4 is extended resulting in Figure 5 that displays the stakeholders for the two market segments: the right and left upper branch are corresponding to the CDP and the walk-up environment, respectively.

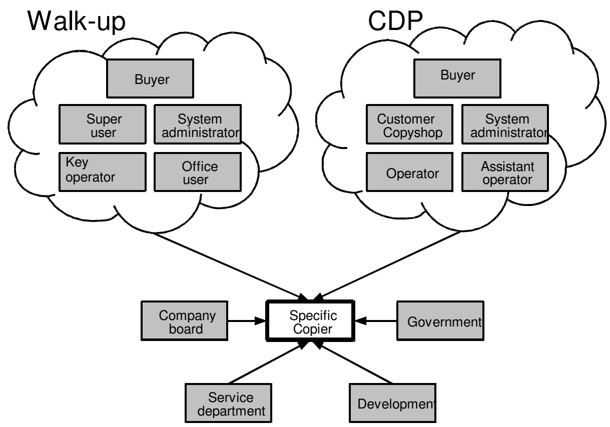


Figure 5. Extended overview of stakeholders

The additional stakeholders for the walk-up copier can be characterized as follows:

- (Generic) Office users: the people that send their documents to the walk-up copier electronically to get them printed or go to the machine themselves to get copies of their documents.
- Super Users (e.g secretaries): experienced and heavy users of the copier. Therefore, often asked by other employees for advice.
- Key Operator: a person responsible to keep the system up-and-running. He knows the copier sufficiently to perform day-to-day maintenance.
- The buyer and system administrator are similar as for the CDP.

To give an idea of how the walk-up environment is used, we describe the **generic office worker** in more detail. This stakeholder represents an office worker who uses the system for (re)production of work documents (typically for his documents, presence of information is more important than appearance).

- He can have any level of education and task in the office.
- He produces mainly standard documents (A4, straightforward finishing) with almost always the same settings. Typically single or a limited number of copies or print-outs. .
- He uses the system both at the control panel and through software on his desktop.
- He works with standard office tools (like windows, office, etc.)
- He knows how to perform his own tasks on the copier, but not more. He asks the super user in case of problems.

Due to the additional market segment for the copier, the process of reasoning, interviewing and discussing was repeated and finally we came up with the following key drivers (note that we kept their numbers restricted, i.e. three for both market segments). For the CDP we selected:

- Productivity (for large volumes): The CDP needs to print large volumes.
- Versatility: suitable for different kind of jobs.
- (Re)production quality: this is what CDP sells.

For the walk-up copier the resulting key drivers are:

- Minimal waiting time: people do not want to wait (long) for their jobs to be complete.
- Ease-of-use: the (inexperienced) user is not interested in the working of the machine at all and prefers a one (green) button approach.
- Reproduction quality.

Note that the integral cost per copy or the related issues of total cost of ownership / running cost have been abandoned as key driver for the CDP to keep the number of key drivers limited. We opted to focus on the actual users of the copier (with the customer of the CDP and the (assistant) operator as the main stakeholders) and less on the buyer who is responsible for the integral cost aspects of the copier. However, integral cost per copy would be added to the key drivers if we would put more emphasis on this type of stakeholder (going to four key drivers for each segment). Typically, the integral cost per copy could be subdivided into application drivers like cost of ownership, costs of consumables, total number of prints (life span), personnel effort (to operate the copier), etc. System requirements related to the application driver consumables are for instance, toner usage, service frequency (e.g. related to number of paper jams) and price, etc.

3.5 Step four: obtain feedback

At several steps during the development of the key driver model, the results were discussed with people from the organization of the copier manufacturer. The people from the marketing and business strategy were viewed as the internal representatives of the customer side as they are closest to them. Their response on the resulting key driver model was enthusiastic, thereby confirming the correctness and value of the model.

3.6 Iterate

In the previous sections only the main part of the whole process of obtaining the key driver model is presented. This already reveals its iterative nature, although many more iterations were made. By iterating over these steps and going from the C via the A to the F view and vice versa many times, a good overview of drivers and requirements was produced.

4. Overview of the key driver model

As mentioned, the copier aimed at two rather different market segments. This means that from both sides we have different key drivers and application drivers that somehow have to be merged into *one* set of system requirements. How this is done is reflected nicely in the key driver model in Figure 7 (at the end of the paper). First, we will give some explanation of the graph:

- Different boxes are used for the system requirements: Rectangles indicate the must-haves, assets that are indispensable for the market. Rounded rectangles indicate unique selling points if compared to the competitors on the market.
- Thickness of arrows: the thicker the arrow, the larger the influence of a specific requirement on the corresponding application driver. For instance, the CDP-application driver print quality is mostly determined by the sheet-image registration.

To explain the obtained key driver model, we highlight some parts. The “minimal waiting time” for walk-up is coupled to the derived application driver “availability.” In general, a walk-up user gets irritated if the copier is not available irrespective of the reason. Thus paper jams have to be very infrequent. The corresponding system requirement is that paper jams have a frequency of occurrence less than a specific number of printed sheets. Another system requirement related to the application driver availability is a large stock of sheets in the paper input module. In this way the general office user is rarely confronted with empty trays. The very low frequency of paper jams and the huge stock of sheets are considered to have a larger influence on the availability than “full and fast error recovery.” Therefore the former ones have thick arrows coming from the application driver, whereas the latter one has only a thin arrow.

There are a number of many-to-many relationships in the graph: the requirement “full and fast error recovery” is related to the application driver “availability” with corresponding key driver “minimal waiting time,” but also directly to the key driver “ease-of-use.” Vice versa, a certain key or application driver is naturally coupled to many requirements.

With respect to the double personality of the copier, some observations can be made. Requirements like “time-to-first-copy” and “time-to-first-print” are typically related to walk-up and not to CDP. This phenomenon is directly related to the different key drivers for both environments. Due to the large volumes and the continuous production of the copier in a CDP environment, the initial waiting time does not have a large influence on the overall speed of a job (productivity/volume). However, for walk-up this has a major impact on the speed of the job (minimal waiting time) due to the low volumes.

The system requirement "cover insertion" is only relevant from a CDP key driver. A typical walk-up user makes only a few copies and inserting a cover can easily be done manually (otherwise he would probably give such a job to the CDP). For producing high volumes as in a CDP automatic cover insertion maintains a high productivity (for various different jobs). The customer objective here is high productivity of versatile jobs.

5. Decomposition linked to requirements

Consider again Figure 3 with the decomposition of the copier into various modules. This decomposition can be mapped on the key driver model, which results in the matrix of Figure 6: the system requirements obtained via the key driver technique are mapped to the subsystems of the decomposition – only the first part of the matrix is displayed for shortness. A "1" in the figure means "related" and an empty cell means "not related."

System Requirements	PIM	REG	FIN	PRINT	SCAN	CONTROL	UI
Scan ahead					1	1	1
First print out	1	1	1	1		1	1
First copy out	1	1	1	1	1	1	1
85 pages per minue	1	1	1	1		1	1
Full speed duplex		1					
Full speed mixed paper (PIM)	1						
56 scans per minute					1	1	

Figure 6. Relating system requirements to system functions

It is clear that a column gives the system requirements that a certain subsystem has to contribute to and consequently this indicates which requirements the "subsystem implementation team" is responsible for. A row indicates how a requirement is distributed over the subsystems and one can see the responsible and contributing subsystems to each individual requirement. This is very useful means for supporting the system design. Next to appointing responsibility, it might also be used as input for budget-based design [10] as one already has a qualitative distribution of a system requirement over subsystems. For instance, time-to-first-print is a shared responsibility of the implementation team for CONTROL, PIM, REG, PRINT, FIN, UI and leads to a rough estimate of the time-to-first-print as the sum of the UI response time, warming-up time (PRINT), and separation / transport / stapling time (PIM, REG, FIN). Note that the duration of the job control and image processing plays a role as well, but good design would aim at performing these tasks (in CONTROL) in parallel with the separation/transport time in PIM and REG.

Of interest are columns or rows that do not contain any numbers at all. A row that does not contain any numbers means that a certain system requirement is not related to any module in the system decomposition and hence, the question arises how this requirement will be met. A column that is empty – which is less occurring in practice – means that a certain module is not related to any of the system requirements and thus not related to any of the key drivers. As a consequence, one might severely question its existence.

6. Strengths, limitations & lessons learned

The strengths of the key driver model are the good (high-level) overview it provides of the system. System requirements are linked directly to the key drivers of the customer (view.) As such, the key driver model is an excellent means to use in discussion and communication. It is valuable for communicating the goals of a project to the stakeholders and moreover, it enhances the understanding between the project leader and his developers. It gives a good means for pointing out why certain requirements are a must and why others are less stringent. A final benefit is its support in making sound tradeoffs between the requirements of a product. Indeed, the model helps tracing the impact of a change in requirements back to the way in which the customer or other stakeholders will perceive the change. After all, satisfying its stakeholders is all a product needs to accomplish.

One of the limitations of the current model is that not all stakeholders have been included. For instance, buyer, service department, and development were not taken into account to keep the focus and the overview. Of course, it is possible to include these in one key driver model (as we briefly discussed for "integral cost per copy" in Section 3.4), but this increases the complexity and possibly more key drivers have to be added. This might cause that the overview and insight are lost. Other key driver models for different stakeholders might be a solution in this case. Building a tool to support the modeling process and visualization (facilitating hopping from one stakeholder to another) would be beneficial. However, the skill is to find a balanced mix of the customers to set-up *one* key driver model that leads to a balanced set of system requirements as in the end only *one* copier is produced. From this broader perspective, the key driver model of Figure 7 would require the extension with key driver "integral cost per click". To keep Figure 7 compact for presentation purposes, we left it out and focused on the actual end users of the machine as discussed before. This is also one of the lessons learned: in order to keep the overview in a key driver model, one has to restrict the number of key drivers, which forces the modelers to make clear choices.

Another lesson learned is that scenario-based reasoning from a customer perspective is very helpful in setting up a key driver model. The relevance of scenarios in this context was also pointed out by [12].

The extension of the key driver technique to include a matrix that maps the system decomposition to the system requirements is an effective means to support the design.

Turning requirements from implicit to explicit has the advantage of clear communication and negotiation. However, the key driver model should be used with the right attitude: in an early design phase requirements might still be up to change and flexibility and openness should be kept.

Constant checking a product against requirements is necessary throughout product development. This might require adaptation of the key driver model. It is a "living model" and that is the way it should be used. However, in general one must aim for stability of the (qualitative) model and aim at changes in only the quantitative part during the design process.

One has to be careful by just copying requirements from previous projects ("re-use of requirements"). However, an advantage of the key driver technique is that, if the requirements for a new product need modification, the technique assist in detecting this. Identifying possibly new key drivers and connecting them to the application drivers and requirements should reveal if (re-used) requirements need to be altered or removed.

7. Conclusions

In this paper we described the key driver technique and presented various guidelines for its use. Its effectiveness was demonstrated by applying it to an existing high-volume copier.

The key driver technique provides guidance to capture the system requirements and to focus the development. The main advantages are the overview it provides, its conceptual simplicity and the clear visibility of the tradeoffs. In this way it forms a good means for communication and discussion, especially since it aims at including the requirements that play an important role and leave out the trivial ones.

Interesting for the specific key driver model presented here is that the copier aimed to fulfill the needs of two different market segments: walk-up and CDP. This was directly reflected in the key drivers that differ for both: minimal waiting time, ease-of-use and reproduction quality for the walk-up and productivity/volume, versatility and reproduction quality for the CDP. The product itself had to unite these needs via *one* set of system requirements.

Several lessons have been learned and we believe that the key driver technique is a useful means to support the design of a copier. Several engineers and architects of the copier manufacturer valued the model. Of course, there is always room for improvement, for instance in requirements elicitation. However, in general we conclude that focusing on the most important issues and providing an overview via a key driver model is a valuable tool in the design of high-tech products like copiers.

References

[1] Muller, Gerrit. CAFCR: A multi-view method for embedded systems architecting; balancing genericity and specificity. http://www.extra.research.philips.com/natlab/sysarch/ThesisBook.pdf, 2004.

[2] A. Antón, J. Dempster, D. Siege: *Managing Use Cases During Goal Driven Requirements Engineering: Challenges Encountered and Lessons Learned,* North Carolina State Univ, TR-99-16, December 1999, http://www.csc.ncsu.edu/faculty/anton/pubs/icse2000.pdf

[3] C. van Lamsweerde: *Goal Oriented Requirements Engineering: A Guided Tour*, Proc. 5th IEEE Int. Sym. Req. Eng. (RE'01), Toronto, August 2001, pp. 249-263.

[4] E. Yu and J. Mylopoulos: *Why Goal Oriented Requirements Engineering,* Proc. 4th Int. Workshop Req. Eng: Foundations of Software Quality, Pisa, June 8-9, 1998, Presses Univ. de Namur, 1998, pp. 15-22.

[5] *Goal Oriented Requirements Language*, http://www.cs.toronto.edu/km/GRL/

[6] Amyot, D. and Mussbacher, G. (2002) URN: Towards a New Standard for the Visual Description of Requirements. 3rd SDL and MSC Workshop (SAM02), Aberystwyth, U.K., June 2002. LNCS 2599, pp. 21-37.

[7] R. J. A. Buhr, Use Case Maps as Architectural Entities for Complex Systems, IEEE Transactions on Software Engineering, Volume 24, Issue 12 (December 1998), pp. 1131 - 1155.

[8] Alistair Cockburn. *Writing effective use cases.* Addison-Wesley, 2000.

[9] Use case maps web page. http://www.usecasemaps.org/index.shtml.

[10] QFD Institute. http://www.qfdi.org.

[11] Freriks, H., Heemels, W.P.M.H., and Muller, G. (2005). On the systematic use of budget-based design. To be presented at INCOSE Intern. Symp. 2006, Orlando.

[12] H. Saiedian, P. Kumarakulasingam, M. Anan. Scenario-based requirements analysis techniques for real-time software systems: a comparative evaluation. Requirements Eng. J, 2005 (10), p. 22-33.

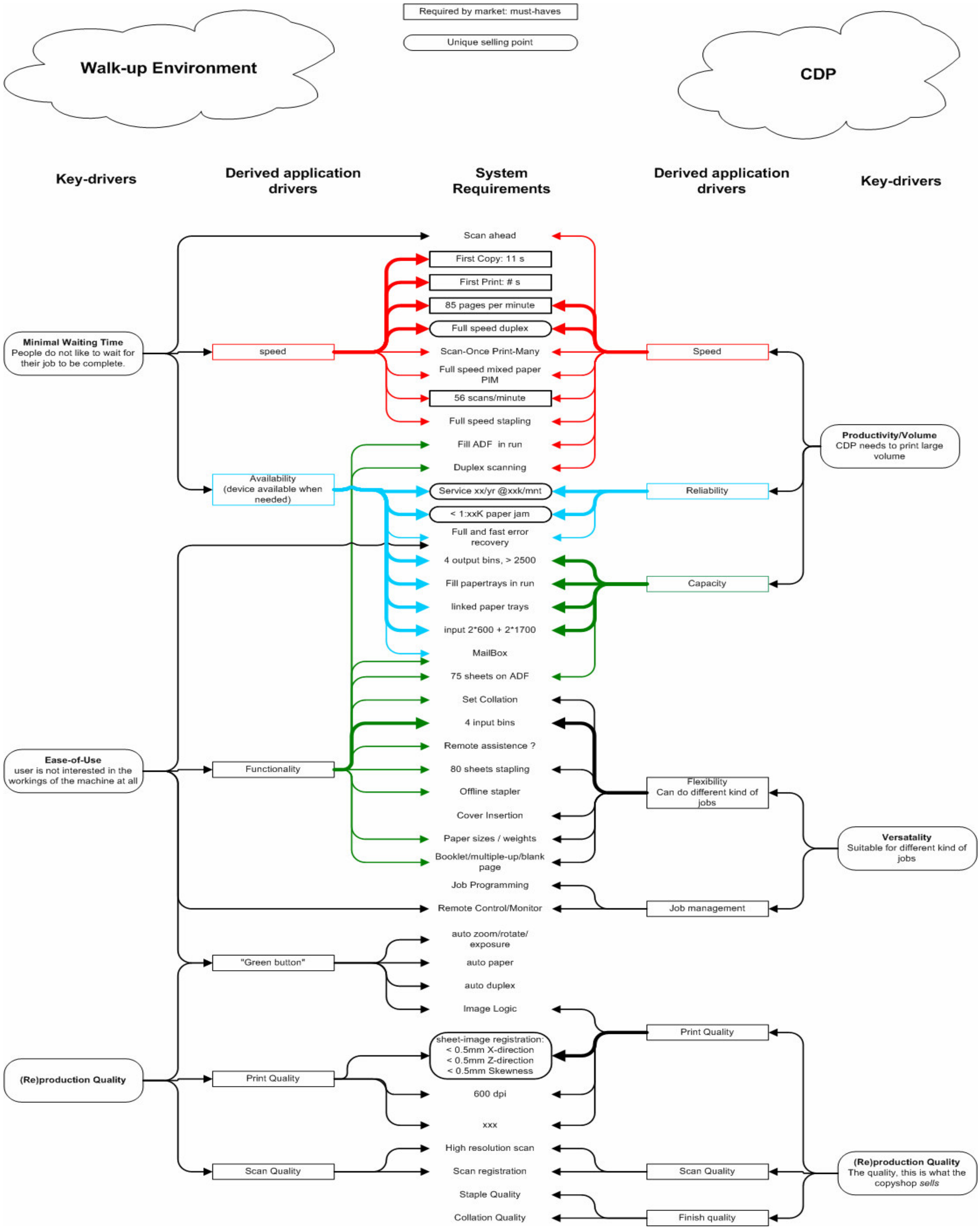


Figure 7. Copier key driver model