Methodic Design of Robot Vision Systems

Joaquin Sitte
School of Software Engineering and Data Communications
Queensland University of Technology
Brisbane, Australia
Email: j.sitte@qut.edu.au

Petra Winzer
Department of Product Safety and Quality
Bergische Universitaet Wuppertal
Wuppertal, Germany
Email: winzer@uni-wuppertal.de

***Abstract*— In this paper we use the design of an innovative on-board vision system for a small commercial minirobot.to demonstrate the application of a demand compliant design (DeCoDe) method. Vision systems are amongst the most complex sensor systems both in nature and in engineering and thus provide an excellent arena for testing design methods. A review of current design methods for mechatronic systems shows that there are no methods that support or require a complete description of the product system. The DeCoDe method is a step towards overcoming this deficiencty. The minirobot robot design is carried from the generic vision system level down to first refinement for a minirobot vision system for visual navigation.**

***Index Terms*— Autonomous robots, robot vision, Design space, Design methodology, Requirement Management, Innovation.**

I. Introduction

Digital image sensors have become small and very cheap. Together with powerful microprocessors and configurable computing devices (FPGAs) we can build sophisticated low cost and compact machine vision systems for widespread use in all types of automation and robotics. Image sensors provide rich information about the state of the world for automation devices and therefore they can be used for very many sensing tasks. It is no coincidence that almost all animals depend crucially on imaging sensors (eyes) for their survival. In primates around one third of the brain cortex is dedicated to vision processing. This indicates that vision systems are among the most complex sensor systems in nature. Likewise the technical use of imaging sensors can be very complex and the field of computer vision has been a fertile research ground for many decades where many advances have been made but many more are still needed to realise the potential of imaging sensors.

The design of complex technical systems is still more art than science, and relies heavily on the knowledge and intuition of the human designer. The design of an artifact always starts with a purpose. The purpose usually implies a diverse set of requirements that the artifact is expected to meet. design is a multi-objective optimization problem because multitude of solutions usually exist that satisfy the individual demands to different degree. The experienced designer carries out the optimization intuitively based on his/her knowledge and experience. Occasionally, a designer may resort to the help of some quantitative optimization technique. Finding an optimal or even only a minimally adequate design solution becomes very difficult when there are more than a handful of design variables because of the exponential increase of the volume of the design space.

In vision systems for robots the interplay between software and electronic and mechanic hardware is quit intricate and the design of a vision system requires the application of many different methods. Traditionally software and hardware are designed by different methods and the results from the application of these diverse methods have to be combined into a single design. which adds to the complexity of the design process and design mistakes are more likely. It becomes difficult to assure that the design objectives are met. A review of design method porposed and in use showed that they address specific aspects but do not support a all aspects of the product [1].

The design methods proposed so far:

1) Originate from the specific design domains of mechanics, electronics or computer software, and are not directly transferable to other domains.. Examples are the VDI Directive 2221 based on the Pahl/Beitz construction method [2], Y-method from circuit design [3]and software engineering methods such as the waterfall method or evolutionary development [4]
2) None of the method specifies a system description, leading to a variety of incomplete descriptions that impede the comparison of the effectiveness of the methods.
3) All the methods specify a process in that they prescribe a series of steps that will lead to the final design. However these steps are not specified clearly enough to allow a unique realisation by different designers and designs..

For (info) mechatronic systems it is necessary to seamlessly handle the mechanic, electronic and software domains. This is true also for robot vision systems where the electronic and software components interact with the optics (focus), mounting (pan and tilt) and possibly a mobile base.

In this paper we investigate the design of a vision system following the Demand Compliant Design (DeCoDe) method developed by the authors [5]. This allows us to integrate the different design methods of the specific disciplines, manage the complexity of the product and the relations between its elements.

The goodness of a design depends directly on the designers grasp of the main demands on the product system and his/her understanding how to best utilize the available components. For example the designer of a vision system will usually be an expert in the field who is expected to have all these aspects in mind. Once such a level of compenetration with the product system has been reached by the designer will not feel the need for a method that makes explicit things that he/she already knows. The problem arises when a new designer has to take over or when a new designer joins the team or when design information has to be communicated to persons outside the team. It is then when DeCoDe method reveals its value because it captures and documents the essentials in an easy to understand form.

This paper is structured as follows. In the next section we give a brief overview of the DeCoDe method that links the three views with the demands on the product system providing a disciplined way of describing, documenting and exploring the design space with reference to machine vision. In section III we build a coarse description of the Robot vision design space using the DeCoDe method. In section IV a detailed design description provides the framework for further refinements and specialisations to demands of specific to a vision system for a small autonomous mobile robot. vision systems. We also briefly describe the actual realisation of the design which is currently subject of detailed evaluation. In section V retrospectively point out the advantages and shortcomings discovered in the design process.

II. Navigating design space with DeCoDe

In any significant design task the designer is faced with a bewildering collection of textual documents that specify requirements, provide guidelines, describe components and materials, and give supplier options, costs etc. We conceived the DeCoDe method to organise this information and facilitate the design process by making it more transparent and traceable, and easier to communicate. This method helps us to describe the hole complex vision system, find out the gaps or the failures of the system design and create new solutions for the vision system. The solutions have to be assessed against the demands for the selection of the best. The DeCoDe method is based on simple ideas for organizing the design information and capturing essential relations between the main design problem elements. It provides a framework for design knowledge documentation and for tracking the evolution of the design. The DeCoDe method is a first step towards a formal description of the design task that will allow the application of quantitative techniques for solving the mutiobjective design problem. One of the main shortcomings of existing design method is that they do not require or assist in obtaining a description of the whole product system.

The hallmark of the DeCoDe is the linking of the demands on the product with three complementary *views* of the product and the maintenance of consistency at all times as the design evolves. These views are the functional view, the structural (components) view and the process view. Each view consists of the corresponding hierarchical list of functions, processes or components. These lists and the demand list are interpreted as tree structured catalogues. *Connectivity* matrices capture the relations between pairs of catalogues and with themselves [6], as shown in Figure 1. There are 10 such matrices.

The rows of a *connectivity* matrix correspond the entries in one catalogue and the columns to the entries in the other catalogue . Each element of a matrix expresses a relation between the corresponding entries in the catalogues. The simplest is a binary value that expresses the existence or not of some interaction. However entries can also be used to indicate the strength of this interaction. The connectivity matrices represent graphs with weighted edges. Consider for example the matrix that relates components among themselves. When we just use binary values for the matrix elements, a one indicating that there is some interaction between a pair of components, and a zero indicating no interaction, the matrix represents a block diagram of the system. This diagram can be enriched in a simple way by rating the strength of the interaction on a scale say from 0 to 10. The matrix now captures more information about the design than the simple block diagram. One can even go a step further and give a measure of influence, which is inherently asymmetric. Component A may strongly influence another component B, while B may not influence A at all. For example the designers choice of the size of any image buffers will be strongly influenced by the number of pixels in an image sensor. But the designer is unlikely to let the size of the image buffer determine the choice of the resolution of the image sensor. In this case the matrix is no longer symmetric.

The matrix that relates the demands to the components can captures some essential information that is not easy to express otherwise. Each row corresponds to a demand and each column to a component. An element of this matrix can represent an estimate of how much a component contributes to the satisfaction of a demand. Thus if a change is made to a component, by checking the values in the corresponding column on can immediately see which demands are affected and even how much. And vice versa, if a demand changes one can read from the matrix which components will be affected.

A. Design space

The notion of design space is useful for exploring design alternatives. Each design solution is a point in the space of design variables. Product design variables are those that influence the satisfaction of the demands on the product. The design space is determined by the product system, and the demands placed on it. Each design solution (design point) will satisfy the demands to a certain degree. Design optimisation searches for the design point that has the highest degree of satisfaction of demands. The solution space consists of those regions in design space that have a high degree of satisfaction of the demands. [1]

For example a digital camera can be characterised by the focal length of the lens assembly, the height, width of the

[1] A different notion of solutions space is used by Gries [7] where the axes are the measures of satisfaction of the individual demands. To find the optimal solution multi-objective optimization is used (Pareto optimal surface).

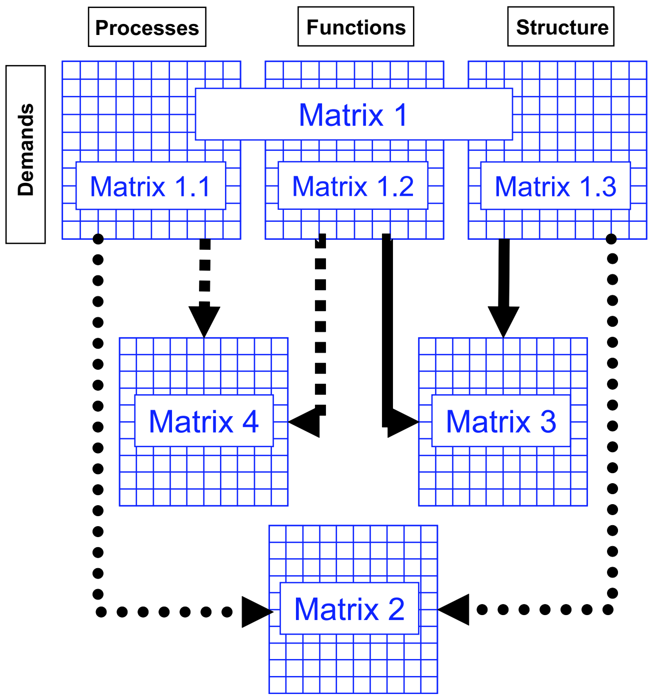

Fig. 1. Diagram of the DeCoDe method. The relations between the demands and the elements of the three main views are described by *connectivity matrices*

pixel elements in the image sensor and the number of pixels in a row and a column of the sensor array. Although this is a highly simplified characterization of digital camera, it may be sufficient for evaluating the image quality produced by the camera. Any digital camera, real or imagined will have specific values of these 5 variables and the designer is free to choose from among a range of possible values. Thus a particular camera can be represented by a point in this 5 dimensional design space. Because a camera is inherently defined by the function of projecting a 3D scene on a 2D image recording medium, there are clearly regions in design space where the corresponding construct no longer meaningfully provides the function expected from a camera. For example a camera focal length of 1 m will still provide the desired function but will be useless in taking a group photo of some friends. Thus although in principle a camera could be anywhere in the design space defined by the range of the variables, useful designs will only occur in particular regions of design space (solution space). These useful regions will be defined by the demands on the product. High magnification tele-cameras will occupy a different region in design space than a pocket camera.

Often variations of an artifact are needed where the requirements vary somewhat around a set of core requirements. In this situations it is useful to develop a reference architecture for the artifact (Example family home, mobile phone camera or robot vision) The reference architecture will list a series of essential components, functions and processes of the artifact and a description of their interrelations so that the resulting artifact meets the core requirements. In the next section we sketch a reference architecture for vision systems. By following the DeCoDe we provide a well defined description of the complex vision system as a hole that provides the basis for our design space.

III. Design of Vision Systems

Application of the DeCoDe method requires the listing of the demands the artifact has to meet, the functions it performs, the components it is made of, and the processes in which it takes part. In this section we list demands, functions, components and processes that apply to any vision system without particularizing on a specific one. These lists provide a reference architecture or initial design template that can be progressively refined for a specific product. The level of detail, or resolution of the lists, is initially coarse, but detailed enough to quickly zero in on the region of design space where the final solution may be found. Once this area has been identified further local refinements of the list will iteratively lead to a solution that meets a specific set of requirements. In this iterative process it may happen that one or more new requirement are uncovered that lead to a completely different region of the design space.

A. Possible demands on a vision system

The demands on a mechatronic system can be roughly divided into three groups: functional, manufacturability and cost, and regulatory and standards. In this paper we focus mainly on the functional demands, that is, the demands that specify the functions the vision system is required to perform. Demands can also be divided into general and specific. General demands are those that any type of vision system must meet. Specific demands arise from a chosen application domain. In this section we will first list general demands and in the next section we will specialise on the specific functional demands arising from a vision system for a small mobile robot.

The purpose of a vision system is to extract from a time varying scene the necessary information on which appropriate actions can be taken. The demand of processing and analysing the time-varying image data distinguishes a vision system from a video camera. Furthermore the processing has to be done on-the-fly as the images are captured. This rules out any *batch* processing of the video sequence at a later time. Figure 2 lists basic demand for a generic vision system.

B. Components of a vision system

Components are the actual physical parts that make up the artifact. For example a basic digital camera will consist of a lens and an electronic image sensor together with some control electronics that allows the captured image to be transferred to another module that will process or store the image. Components are designed to perform one or more specific functions within the artifact. Components themselves will be made up of parts that are also components at the next lower level. In this way components typically form part of

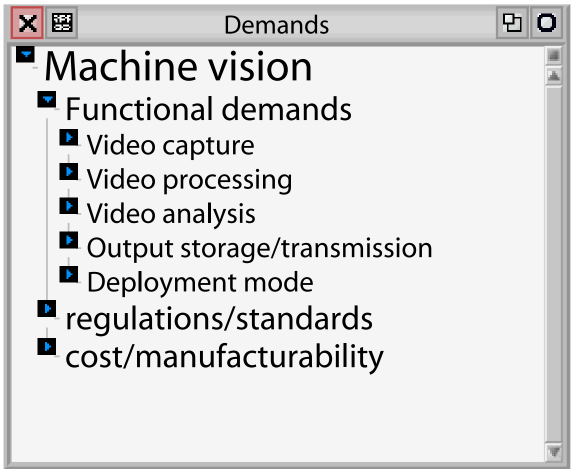

Fig. 2. Demands for a generic vision system.

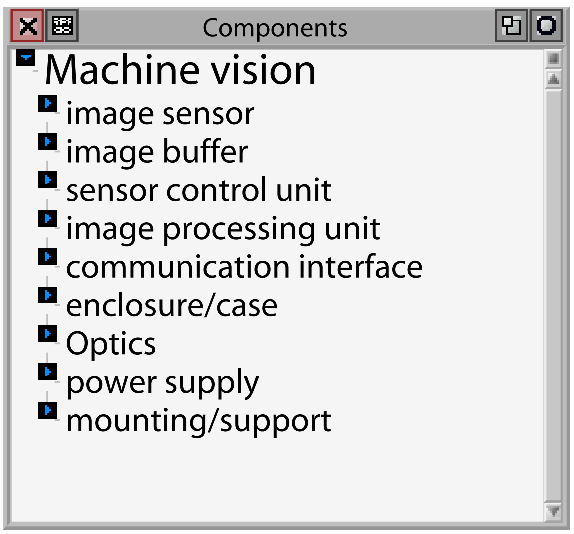

Fig. 3. Componens of a generic vision system.

a hierarchical component tree. This hierarchical component structure is closely related to the a view of the artifact that organizes the artifact in logical subsystems, subsubsystems and so forth.

Figure 3 lists the components any vision system will comprise.

By constructing an influence matrix between the components, as described earlier in this section it is possible to get a first estimate of the most and least critical components in the design as illustrated in Figure 4

C. Functions of a vision system

The way an artifact meets its requirement is by performing one or more functions, such as capturing an image and the transferring the image to a storage medium. Functions are closely related to the demands, but here is not always a one-to-one correspondence. Demands not only imply a function but also qualify and constrain that function. A function is independent of any specific realization of the artifact or

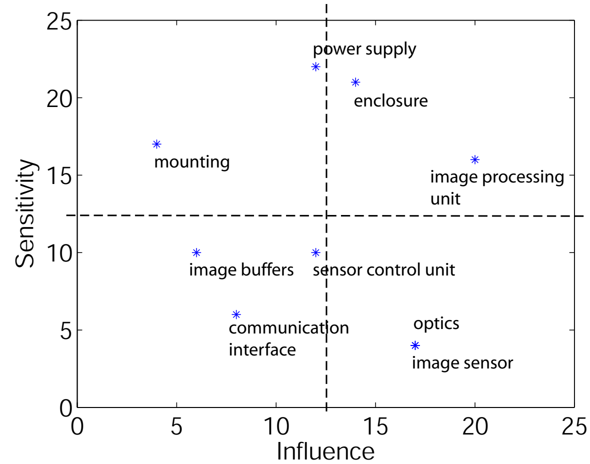

Fig. 4. Mutual influence of mobile robot components. The components in the upper left quadrant are critical in the sense that have strong influence but also are highly susceptible to other components

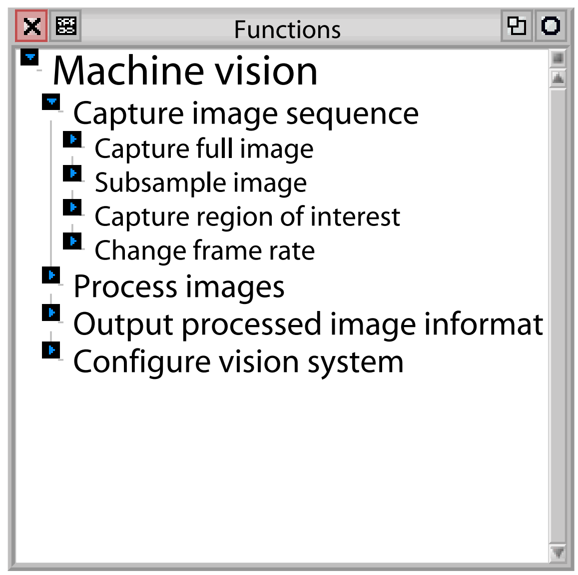

Fig. 5. Functions of a generic vision system.

its components. Usually there are many ways to realize a function with different combinations of components. Function 5 give the main functions of a generic vision system.

D. Processes in a vision system

In order to perform a function one or more components of the artifact need to carry out one or more processes. In a process the components undergo a dynamic interaction. For example the image capture function is realized integrating the photocurrent at each pixel element over a short amount of time. The photocurrent is produced by the light projected by the lens onto the image sensor. This process is repeated whenever an image is captured. It is important to understand that processes are changes of system variables that occur during a finite time interval. Processes start, run and stop.

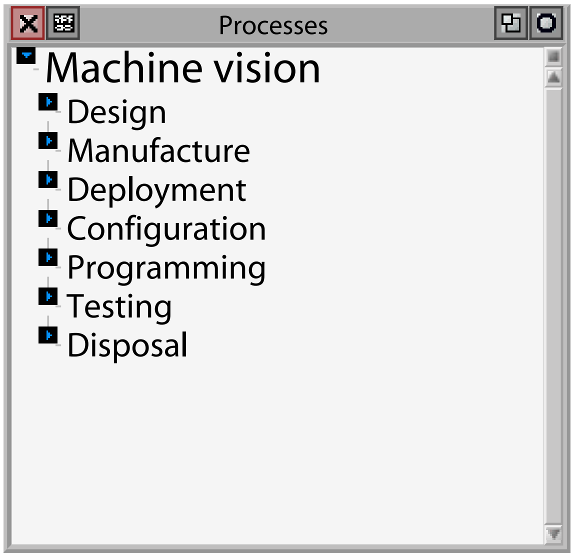

Fig. 6. Top level processes of in a vision system

Only a few special processes go on forever. Processes need physical components to run on.

A design is fully specified when all the functions have been described, all the components have been described and all the processes have been defined. Functions, components and processes must be consistent, that is the components must be such that they can execute the processes required to realize a specified function. Figure 6 give the main functions of a generic vision system.

IV. Small mobile robot vision system

In this section we describe how the generic vision system description was refined following the DeCoDe to arrive at an innovative vision system design for the Khepera mini robot [8]. A low cost visible light image sensor is to be the main sensor that provides the robot with information on spatial structure in the robots environment so that it can navigate autonomously. This means that it must be able to avoid obstacles while moving towards a target destination following an appropriate path. Paths are not specially marked for the robot, rather it must determine a path to the target that is suitable to its terrain going capabilities. The camera data stream from the sensor must be processed fast enough to allow the robot to respond in real time to changes in the environment and its own motion. To maintain autonomy the video data stream must be processed of onboard. From this general statement of demands and the constraints imposed by the existing Khepera robot specific demands can be derived. These are shown in figure 7.

The requirement of onboard image processing was made after discarding the option of off-robot image processing as incompatible with the other demands. Although real time image processing could be realized on a remote processor provided that there is sufficient communication speed t transmit the raw images. Wireless transmission is the only alternative

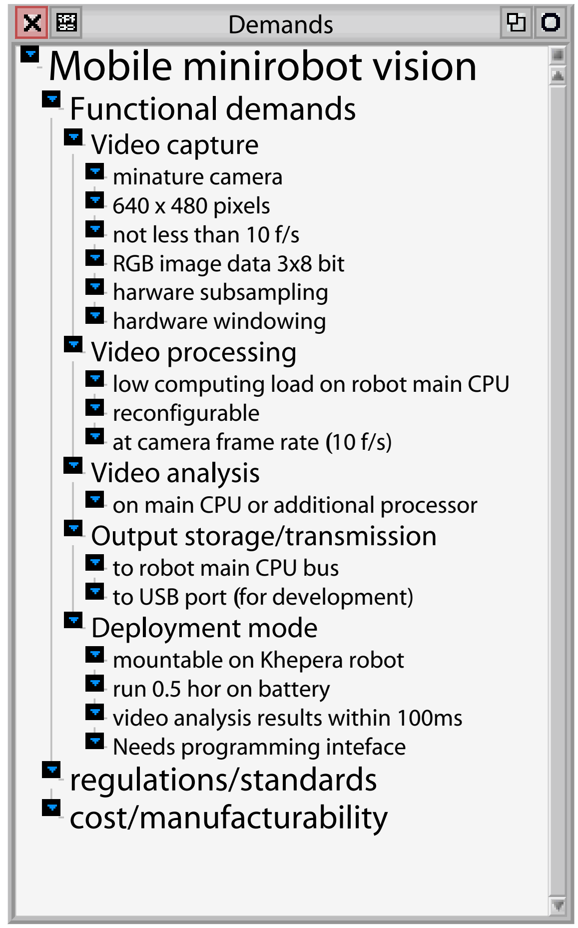

Fig. 7. Demands on the mini robot vision system

because the requirement for mobility precludes any wired link to the robot. Without further analysis it is safe to say that the required data rates can not be supported by current wireless technology compatible with the other demands on power consumption, size and cost.

For the robot vision system the image analysis function consists in providing the environmental information necessary for the robots navigation. This in turn implies The robot must distinguish where there is free space to move. Free space means the is a suitable surface on which it can move. the free surface may have texture, markings or shadows that might resemble objects or voids. No-free space is where there are depressions or holes in the ground or elevations that the robot can not overcome.

The demands are of two types those which specify an unrestricted capability and those that specify constraints that affect those capabilities. For example the ability of recognizing free space is unrestricted as we would like the robot to be able to do this regardless of the nature of the structure of the environment. However there may be situations where this requires an amount of data processing that the robot cannot perform in the available time with the available resources. Therefore the demand of recognizing free space is constrained

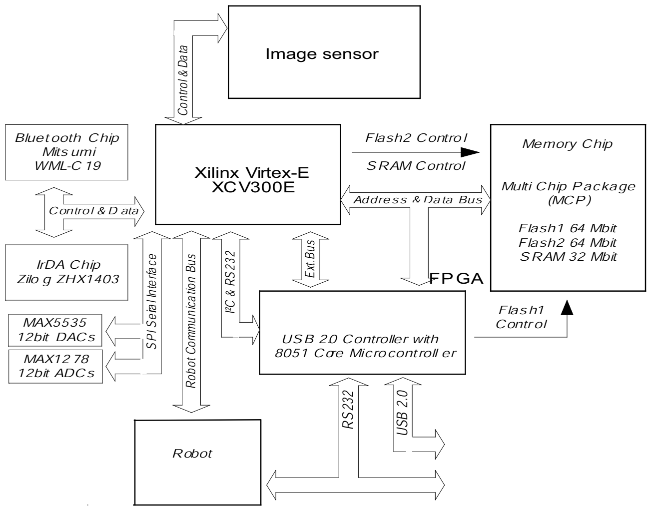

Fig. 8. Block diagram of mobile robot vision module.

Fig. 9. Vision system attached to the top of the minirobot Khepera (courtesy of SCT-University of Paderborn)

by the other demand that it has to be done with the specified resources in a specified maximum time.

These demands were ordered by their relative importance using the relation $v(a) - v(b)$ on a scale of $[-10, 10]$ where a and b are demands and $v(x)$ represents a measure of importance. By estimating the influence and dependence between all pairs it is also possible to cluster the requirements.

A. Functions of the mobile robot vision system

The functions of the vision system for a small mobile robot mainly differ in the image analysis part from those listed in Figure 5. The image processing function for the mobile robot is: Determine free space in the space covered by the field of view of the camera within a time that allows the robot to maneuver at the robots maximum speed without leaving free space. The description of free space means finding the direction and distance of the boundary points of free space. How many of these points can be determined depends on the algorithm and the computation speed of the vision system.

B. Components of the mobile robot vision system

The expanded list of components for a mobil robot vision system is too long to be reproduced here. Instead Figure 8 shows a simplified block diagram of a solution intended to meet the requirements outlined above. Figure 9shows the actual hardware built according to the design [9].

The capabilities of a vision system depend to a high degree on software. The software acts as buffer in the propagation of the requirement down to underlying hardware. Many different algorithms will run on the same hardware as long as a minimum of services are provided by the hardware.

V. Design Summary and Conclusions

The SCT-FPGA board based design fulfils the main requirement of being able to carry out significant image processing on the robot while also fulfilling the demand of small size and low power consumption. The configurability provided by the use of an FPGA as the main image processing component takes this vision system to new levels of performance when comapared with the CMUCam [10] who is the standard of reference for small low cost robot vision systems. Although the evaluation SCT-FPGA vision system is still ongoing its suitability for simple autonomous navigation tasks has been demonstrated.

When initially proposed the DeCoDe methos seemed to require too much extra work from the designer. The application of the method in an industrial problem described briefly in [1]suggested that it may not be necessary to carry out the method in its full detail to obtain benefit of the time savings obtained by designing a better quality from the start.

References

[1] J. Sitte and P. Winzer, “Evaluation of a new complex system design method on an mechatronic automotive product,” in *Proceedings of the 2006 IEEE International Engineering Management Conference (IEMC 2006)*. Salvador, Brasil: IEEE, Sept. 2006, p. .

[2] G. Pahl, W. Beitz, J. Feldhusen, and K. H. Grote, *Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung*, 7th ed. Berlin, Germany: Springer, 2006.

[3] D. D. Gajski and K. R. H., “New VLSI tools,” *IEEE Computer*, vol. 16, no. 12, p. 11 14, Dec. 1983.

[4] I. Somerville, *Software Engineering*, 8th ed. Pearson Education, 2007.

[5] J. Sitte and P. Winzer, “Mastering complexity in robot design,” in *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2004*. Sendai, Japan: IEEE, Sept. 2004, p. 1815 1819.

[6] A. Lex, “Methodik zum Anforderungsgerechten Roboter Design,” Master’s thesis, Bergische Universitaet Wuppertal, Wuppertal, Germany, 2003.

[7] M. Gries, “Methods for evaluating and covering the design space during early design development,” Electronics Research Lab, University of California at Berkeley, Tech. Rep. UCB/ERL M03/32, Aug. 2003.

[8] K.-T. SA, *Khepera User Manual, version 4.06*, Lausanne, Switzerland, 1995.

[9] T. Chinapirom and U. Witkowski, “2d camera module for autonomous robot,” System and Circuit Technology, University of Paderborn, Tech. Rep., 2005.

[10] A. Rowe, C. Rosenberg, and I. Nourbakhsh, “A second generation low cost embedded color vision system,” in *Embedded Computer Vision Workshop, IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.