ITCS 2005

Hans Weghorn (Ed.)

Information Technology & Computer Science

Proceedings of the
2nd Annual Meeting on
Information Technology & Computer Science
at the BA-University of Cooperative Education

Stuttgart, Germany, May 2005

BA

Volume Editor

Hans Weghorn
Information Technology & Applied Computer Science
BA-University of Cooperative Education, Stuttgart
Germany

Proceedings of the 2nd Annual Meeting on
Information Technology and Computer Science – ITCS 2005
Stuttgart, Germany, May 2005
Hans Weghorn (Ed.)

Printed in Stuttgart, Germany

ISSN 1614-2519

Preface

This year, the second annual meeting on Information Technology and Computer Science of the Berufsakademie Baden-Württemberg (BA-University of Cooperative Education) offers a full day of interesting presentations. These contributions are provided from five different BA-University sites. The first section deals with embedded systems and its recent technologies, while the next section addresses the exciting topic of signal processing in application to image and speech data. In addition to these clearly categorizable papers, in the last section there are contributions about other amazing topics, e.g., robotic soccer and emotion recognition, and also some standard topics out of the IT area.

Development and research work usually is conducted at the BA in the frame of student projects, which are part of the regular studying course and which are guided by Professors and other teaching personal. This applies to most of the papers included here. On the other hand, a new possibility of elaborated research work is enabled by collaboration with one of our partner Universities – the highly reputed School of Computing of Staffordshire University – that even allows us to supervise Ph.D. work at the BA. Some of the here presented projects belong to this category, and the implementation of this new kind of very advanced education and research certainly represents an important strategic topic for the future of the BA.

I want to thank all authors for their working invests in our second ITCS meeting. Furthermore, I want to thank all attendees for their interest in this event, and I wish all an informative day of presentations. Finally, I like to express my hope and vision that this meeting in future will annually move between the different BA sites, since we have available a lot of attractive places for our BAs in the nice southern part of Germany that would be worth being visited.

Meeting chair – ITCS 2005

Hans Weghorn

Table of Contents

Section on
Embedded Computing

Web-based Services in Measurement and Control with Embedded Linux

Joachim Schmid

Berufsakademie Karlsruhe – University of Cooperative Education, Information Technology
Erzbergerstr. 121, 76133 Karlsruhe, Germany
Joachim.Schmid@BertholdTech.com

1 Introduction

Web based services are used more and more in measurement and control. They provide visualisation of measurement data via web browsers, and give the ability to configure and manage the device using the Internet. Another point is the automatic e-mail notification in cases of error. Newest approaches also try to realise distributed automation architectures utilizing Web services.

The basic requirement to use Web based services, like HTTP, FTP, SMTP and other techniques, is a TCP/IP-protocol stack. But the implementation of such an IP-stack is very challenging and needs a lot of resources, which will not necessarily be available in embedded systems. Another possibility to get an IP stack is to integrate an additional hardware chip, which provides an Ethernet interface, an implementation of an IP stack, and other software components. But this solution depends strongly on the device manufacturer (e.g. Any-Bus-IC from HMS).

Alternatively, more complex operating systems like Windows CE or embedded Linux can also be used for obtaining access to a TCP/IP-stack, but this solution often means the use of additional hardware.

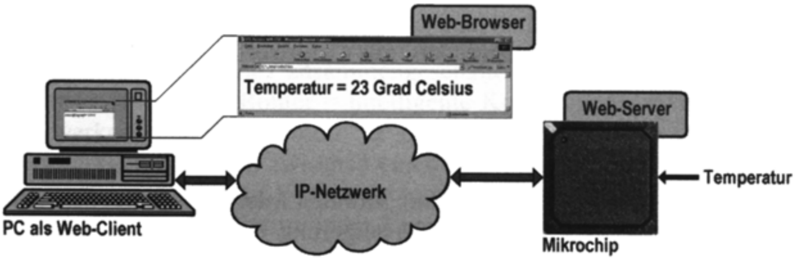


Fig. 1: Visualization of data measurement access via a web browser.

2 Embedded Linux

For the use of embedded Linux there exist several good reasons:

- There are many low-cost hardware solutions available

- The software is freely available (GNU public license), it is and maintained by a vast community of developers
- Transparency of source code (Open Source)
- There is a C(++) compiler toolchain for crossplatform development (also for free)
- Libraries are optimized for embedded systems (uClibC, Busybox)
- Real-time ability, e.g. with RTAI-extension
- Scalable (complete Linux system uses less than 2MB flash memory)
- Extensive support of hardware interfaces
- Security (firewall, operating established, etc.)

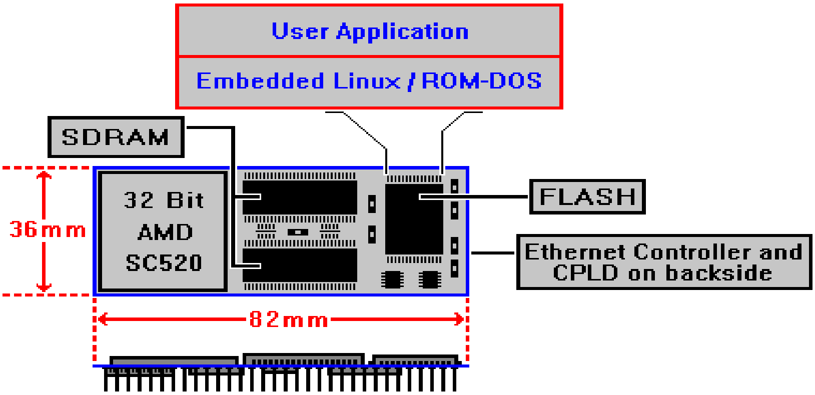


Fig. 2 Small hardware solution suitable for embedded Linux.

3 Conclusion and future development

Beside more capacity and power, recent generations of micro controllers provide the integration of Ethernet interfaces at very low costs (e.g. the TRICORE product line from the company Infineon [10]). Because of these advantages in performance, it is presumed that in future embedded computing complex operating systems will be used more often. This is also due to security issues, which plays a big role in industrial applications.

References

1. Walter, K.-D.: Embedded Internet in der Industrieautomation (2004)
2. Walter, K.-D.: Messen, Steuern, Regeln per Internet (2003)
3. PRAXIS Profiline: Industrial Ethernet (2004)
4. Any-Bus-IC: http://www.hms-networks.com
5. GNU Cross toolchain: http://www.gnu.org
6. HollaBaugh, C: Embedded Linux, Hardware, Software, and Interfacing (2002)
7. uClibC: http://www.uclibc.org
8. Busybox: http://www.busybox.net
9. RTAI (Real Time Linux): www.rtai.org
10. Tricore-microcontrollers: http://www.infineon.com/tricore/

Modular Interface Driver for Automotive Electronic Control Module Testing

Konrad Reif[1], Michael Becker[2], Thomas Krehain[2], and Florian Schwab[2]

[1] Berufsakademie Ravensburg, Fallenbrunnen 2, 88045 Friedrichshafen, Germany
reif@ba-ravensburg.de
[2] Audi AG, 85045 Ingolstadt, Germany
{Michael.Becker, Thomas.Krehain, Florian.Schwab2}@audi.de

Abstract. A universal, modular interface driver was developed in Windows XP, as a dynamic link library in C++ to ensure versatile diagnostic communication. This interface driver is used for an automated electronic control module test station at AUDI AG.

1 Introduction

Car diagnostics is not only a valuable and indispensable tool used at the stages of initiation, commissioning and maintenance, but also in development. In comparison to production and service applications, development needs more degrees of freedom in car diagnostics. The software for electronic control units (ECUs) is usually developed by the ECU Manufacturer. During its development the software is continually changed and improved. Therefore various tests at an automated test station are necessary. The interface driver described in the present article is used for this kind of tests. It allows interactive communication between the ECUs and a PC which is used as an automated test station.

2 Basics

A protocol, as understood in this context, is a control installation to safeguard communication between two different participants arranging for unique syntax, semantics, pragmatics and time cycles. An interface driver is a program routine enabling the exchange of data between two partners of communication. This can refer to the administration of a hardware element parameter or to the control of syntax, semantics and pragmatics at the transition interface between different protocols. A dynamic link library (dll) is a memory saving variant of a link library. Whereas a static link library transfers the source code of any call into the application, a dynamic link library uses dedicated links. This means that a dynamic link library loads the code into the RAM just once, thus saving a large amount of memory and data space.

At present, the CAN bus is the standard used in the automobile industry. It is used for AUDI AG products in different forms and varying speeds: e.g. high-speed CAN (500 kBit/s) and low speed CAN (100 kBit/s). For power train applications, cluster modules and diagnostics high speed CAN is used, whereas comfort systems and infotainment applies low speed CAN.

ECUs are capable of on board diagnostics to discover and classify causes of errors occurring in the car. In this context, electrical and electronic components are tested in the car, and often during drive operation. If errors occur, they are recorded in the ECU's fault code memory involved and even shown by a MMI system, if necessary (Man Machine Interface, e.g. the cluster module). At initiation, commissioning and service, but also in the development stage, there is also off board diagnostics used. It is using the exchange of information between the ECU's and the so called diagnostics tester, to localize errors in the vehicle. Furthermore certain actuators in the vehicle can be remote controlled or software updates can be performed. The data exchange between the test station and the vehicle takes place at current vehicle models at the AUDI AG via diagnostics CAN. It connects the standardized diagnostics interface with the gateway module.

The TP 2.0 protocol establishes a link between exactly two participants for transmission of large amount of data. This link is called diagnostic channel. The protocol offers the possibility to open more than one diagnostic channel. It contains arrangements about start-up and break-up of the diagnostic channels, about data transmission, timing cycles and the link control. The Key Word Protocol 2000 is a communication protocol for diagnostic data defined along international standards. At AUDI AG a slightly modified version is used. Here the request-response approach is used, which is characterized by sending a response on any request from the diagnostics partner. The protocol describes the sequence relative to the request of diagnostics data and the adequate response with the relevant data or fault codes issued

3 Concept

The interface driver has to meet the following requirements: allow coding of diagnostics communication and tests based on C++. Therefore, both diagnostics tester and diagnostic simulation of an ECU have to be implemented. The interface driver has to be modular consisting of different diagnostics layers, so that obsolete protocols can be easily replaced by new ones. In this way reusability can be ensured.

The application, e.g. a diagnostics service, accesses only the KWP 2000 layer and receives relevant responses. Then the KWP 2000 layer accesses once more the intermediate layer between KWP 2000 and TP 2.0, and so on. This causes data flow through all driver layers, and completes the protocol specific syntax in all layers and executes the relevant program routines. The application accesses the driver via an individual class. The KWP 2000 services implemented provide the member functions of this class. Only the class is to be imported to link the interface driver with an application. This is a considerable simplification as compared to an individual declaration of any driver function of its own. Furthermore all necessary initialization processes can be carried out in the constructor of each class, and any de-initialization can be carried out by the destructor.

Is a diagnostics service being called during an application, the interface driver opens a channel to the target ECU and sends a relevant diagnostics command. The target addresses of the ECUs are agreed upon and documented in the corporate group specification. If the same ECU is being accessed by a second diagnostics service, the current diagnostics channel is retained. A second channel may not be initiated yet, if the first diagnostic channel is still being processed. If a diagnostic service is addressing another ECU, the current diagnostics channel will be closed and a new one opened. The current diagnostics channel will also be closed, if the specific class destructor is accessed.

The drivers' responses are ensured via callback functions. These callback functions are custom designed response routines, in which the relevant function addresses the calling function about a certain event. The callback function is initiated in the accessed driver layer; however, it is implemented in the calling layer. The callback function holds contains the received application data and evaluation parameters as reference values. These parameters disclose the length of data, their segmentation, if they are part of a positive or a negative response to a request, or if a fault has affected the driver itself.

4 Implementation

Any interface driver layer and intermediate layer were implemented on a commercially available PC as a specific dynamic link library in C++ and under Windows XP. Here just a short overview is provided, for more details see [1,2,3]. The CAN hardware interface and Vector Informatik GmbH [4] drivers were used as basic modules. Based on this hardware, a CAN-TP 2.0-intermediate layer was implemented as a callback function. The TP 2.0 layer is arranged in two C++ files. The interface driver accesses internally the TP 2.0 layer via a specific class. The source code of this data file represents the TP2.0 layer link to superior layers.

The KWP-2000-TP-2.0-intermediate layer is implemented as its own C++ data file. It enables the communication between the KWP 2000 layer and the TP 2.0 layer. It simultaneously processes the KWP 2000 layer request to send data, initializes the KWP 2000 layer, addresses the TP 2.0 function as specified in the protocol command, and returns received data or fault codes via callback functions. The KWP 2000 driver layer is also implemented in a C++ data file. It is to apply the necessary KWP 2000 functions and to transmit the data input to the user application as required.

Should any error occur during communication between the individual layers or among the program routines, an error handling mode is provided. This informs the user on the error type occurred in which layer. The callback function ensures the necessary feedback to the user of the system. This function transfers a specific status parameter indicating an error as recognized indicating and also the code of the error identified. The subsequent steps will not be executed any longer, if an error occurs while processing any of the routines, and the diagnostics channel will then be closed, if needed.

5 Conclusions

A modular interface driver was developed, which allows its implementation in KWP 2000 services, as required for an automated ECU test station. Furthermore it creates the possibility for ECU simulation. Should any errors occur, their causes can be retrieved within the driver. This approach allowed programming of diagnostics communication between a specific test application and an ECU in C++. The source code of the interface driver is available to the students of the University of Cooperative Education (BA Ravensburg) for further students' research projects and examination theses, and it is continuously improved and further developed. Current and future projects lead to enhancements of the KWP 2000 layer, further development of diagnostics application with a graphical user interface, and the extension to multi channel access. Moreover the initiation of ISO TP (ISO 15765-2), UDS (ISO 14229-1) and advanced functional addressing of ECUs are in work.

References

1. M. Becker, F. Schwab, Umsetzung des VW CAN-Transportprotokolls TP 2.0 auf Basis des Betriebssystems Windows XP, Studienarbeit, BA Ravensburg, 2004
2. T. Krehain, Realisierung eines universellen Schnittstellentreibers für einen automatisierten Wegfahrsperren-/Komponentenschutz-Prüfplatz, Diplomarbeit, BA Ravensburg, 2004
3. M. Becker, T. Krehain, K. Reif, F. Schwab, „Schnittstellentreiber für einen automatisierten Steuergeräte-Prüfplatz", special issue „Automotive Electronics" of the journal ATZ Automobiltechnische Zeitschrift, vol. I/2005
4. Vector Informatik GmbH, Getting Started with the CAN Driver Library, http://www.vektor-informatik.de

IPsec for Embedded Systems

Sabrina Riedmüller[1], Uwe Brecht[2], and Axel Sikora[3]

[1] Bucher Hydraulics GmbH, 79771 Klettgau, Germany
s_riedmueller@web.de
[2] MIMOT GmbH, SMD placement, 79541 Lörrach, Germany
brecht.uwe@gmx.de
[3] University of Cooperative Education, 79539 Lörrach, Germany
sikora@ba-loerrach.de

Abstract. The growing number of embedded devices with interconnection to the Internet causes severe security risks. VPN-oriented countermeasures suffer from the fact that embedded devices come with only very limited resources. This contribution discusses the requirements for an embedded VPN based on IPsec. IPsec is an extension of the IP protocol to encrypt and to authenticate IP packets for secure transmission.

1 Introduction

1.1 Embedded security

The growing number of interconnected microcontrollers in technical systems causes several security risks. Manipulation of a point-of-sale (POS), an alarm system or an industrial plant can cause huge and costly damage [4]. Virtual Private Networking (VPN) allows authentication and encryption of data, and thus is a major element to provide security for IP-based systems. However, only basic approaches exist for embedded systems [1,2]. This paper reports about a real implementation of IPsec in an embedded TCP/IP stack (http://www.embetter.de).

1.2 VPN

To minimize the security risks virtual private networking (VPN) has been developed. A tunnel between two nodes allows encryption and authorization and secures the transmission of data over the public Internet. VPNs may be realized on different levels of the TCP/IP protocol stack [7]:

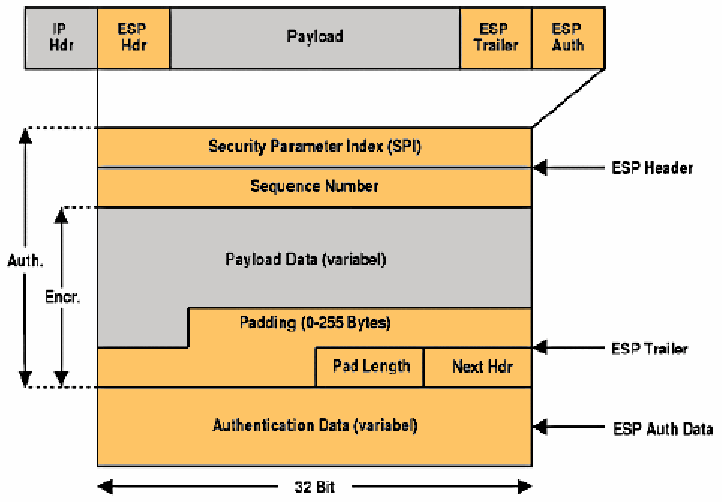
Layer	Protocol
2	L2TP, PPTP, L2F, MPLS
3	IPsec, GRE
6 and 7	HTTPS, SSL, TLS, PGP, S/MIME

1.3 IPsec

IPsec is an extension of the Internet Protocol, which is described in RFCs 2401 to 2412, with further RFCs being published. It offers a framework for security that operates at network layer; therefore, it is application independent. IPsec supports two modes, the transport and the tunnel mode. Transport mode encrypts only the data part of each packet, but leaves the header unchanged. The tunnel mode provides a higher level of security because both data and header are encrypted. Each mode offers three possibilities for sending data, i.e. authorized, encrypted, or both methods together. For each transmission method, there are several possibilities for the key exchange. Creating a security association (SA) is also possible in two different ways. All these varieties of options make an implementation of IPsec very complex but also very flexible. For the use in embedded systems encryption is considered at least an approach securing confidentiality and authentication. This leads to the use of IPsec with Encapsulated Security Payload (ESP) in tunnel mode.

1.4 Encapsulated Security Payload (ESP)

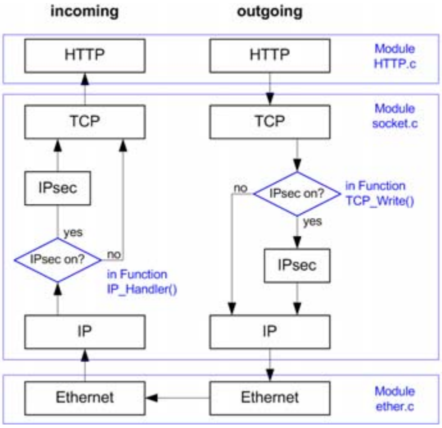
The ESP Header provides authentication, integrity checking and replay protection [9].



2 Implementation of IPsec

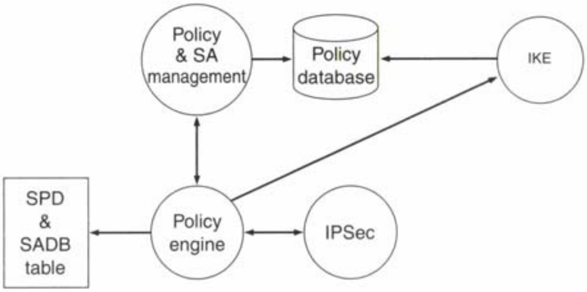
2.1 IPsec in an embedded TCP/IP stack

The following Figure shows the basic handling for the IPsec frames in a TCP/IP-stack:



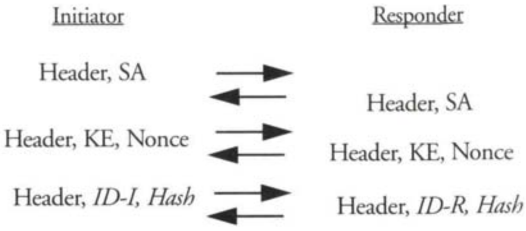
2.2 IPsec Architecture

IPsec consists of databases and management engines which handle encrypted messages. The policy engine works like a firewall script, i.e. if there is no rule, the packet must be discarded. In the security association database (SADB) the encryption algorithms, authentification algorithms etc. for each connection are stored. Internet Key Exchange (IKE) is responsible for the key management and based on ISAKMP, which establishes a security association (SA). For each SA is an entry in the IPSec SADB applied [5,6,7,8].

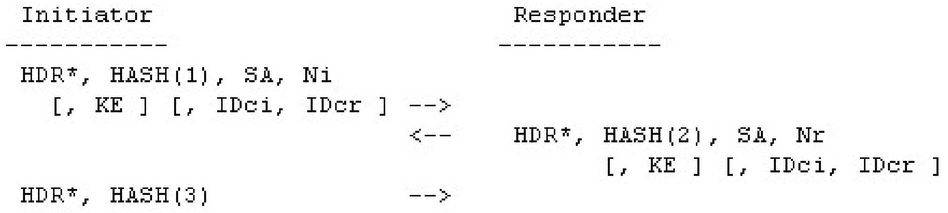


2.3 Creating an IPsec SA

For establishing an IPsec SADB entry are two different phases necessary. In the first phase the two devices transmit 6 packets to create an ISAKMP SA.



In the second phase the two devices send 3 encrypted messages to create the IPsec SA.



```
 Initiator                        Responder
-----------                      -----------
 HDR*, HASH(1), SA, Ni
   [, KE ] [, IDci, IDcr ] -->
                           <--    HDR*, HASH(2), SA, Nr
                                        [, KE ] [, IDci, IDcr ]
 HDR*, HASH(3)             -->
```

2.4 Algorithms

One of the main challenges in providing embedded VPN is the implementation of the computing and memory intense cryptographic algorithms. There are only few implementations known on the market. The WAKAN toolkit is one of them. It is available to the students of UCE Loerrach through the activities of a local Steinbeis Transfer Centre (http://www.stzedn.de).

2.5 Memory usage

The memory usage of IPsec is a challenge for low-end embedded devices, however an implementation is far from being unrealistic. Storing 3 SAs requires some 1 kByte RAM. The whole implementation needs about 8 kByte RAM and 64 kByte data memory. The crypto algorithms and the handling of large numbers take the largest portion out of it. A reduction of memory would be possible, if only DES/3DES would be implemented. However, as those don’t allow a significant security level, the described solution includes AES, as well.

3 Conclusion

IPsec offers very detailed solutions for many use-cases. The complexity makes an implementation quite awesome. However this contribution shows that it is well feasible to run IPsec on mid-size microcontrollers.

References

1. Straub, T., Sikora, A., "Cryptography on Embedded Systems", Proc. Embedded World Conference 2005, Nuremberg, S. 517-536.
2. Wollinger, T., Guajardo, J., Paar, C., “Cryptography in Embedded Systems: An Overview”, Proc. Embedded World Conference 2003, Nuremberg.
3. Menezes, A., van Oorschot, P., and Vanstone, S.. Handbook of Applied Cryptography, CRC Press, 1997.

4. IAONA e.V. The IAONA Handbook for Network Security. Draft/RFC 0.4. Available at http://www.iaona.org.
5. Doraswamy, N., Harkins, D., “IPSec”, Prentice Hall PTR, 2003.
6. Kaufman, C., Perlman, R., Speciner, M., “Network Security”, Prentice Hall, 2002
7. Sikora, A., “Technische Grundlagen der Rechnerkommunikation”, Fachbuchverlag Leipzig, 2003.
8. Raepple, M., “Sicherheitskonzepte für das Internet”, dpunkt.verlag GmbH, 2001
9. http://www-rnks.informatik.tu-cottbus.de/de/teachings/2003/WS/Proseminar ...
...ComputerNetworking/materials/SebastianPuder.pdf

Section on Signal Processing Applications

Astronomical Image Compression by Segmentation and Modeling of Image Dynamics

Christian Grünler[1,2], Hans Weghorn[1], and Claude Chibelushi[2]

[1] BA-University of Cooperative Education, Faculty of Information Technology,
Rotebühlplatz 41, Postfach 10 52 80,
D-70178 Stuttgart, Germany
[2] Staffordshire University, Faculty of Computing, Engineering and Technology,
ST18 ODF Stafford, United Kingdom
c.d.grunler@staffs.ac.uk

Abstract. Sky surveys and virtual observatories provide astronomic data at datarates that exceed available storage capacities. Only compression of the vast data volumes will enable effective scientific exploration of the generated data. A computationally efficient method is proposed in this paper.

1 Introduction

Virtual observatories (VOs), which provide high-quality astronomy data, will make it possible for astronomers to access digital data instead of waiting until access to a telescope is granted [1]. Valuable data will be accessible to more astronomers as sources of different data sets will be linked. VOs need huge data processing and storage capacity for several reasons: the number of large telescope facilities has grown dramatically [7], the resolution of imaging devices is increasing steadily [3], and astronomical projects nowadays involve observations of several wavebands. Furthermore, survey telescopes, which are currently being planned, will image the entire sky continuously every night and therefore generate Petabytes of image data. An international VO will only become a success, if the problems of limited storage capacity and bandwidth can be solved. The research work described here targets this problem by compressing the raw images stored in astronomical archives.

2 Previous Work: Methods for Lossy and Lossless Compression

Several approaches for lossy compression of astronomical images using transform-based methods exist [13],[6],[5], but lossless compression techniques, which preserve astrometric and photometric attributes, are needed to encode precious primary data. Generally, it has to be possible to verify the results of investigations on undegraded data, especially if the results present new and revolutionary findings. Lossless compression tools, for astronomical data are quite rare. Some transform-based lossy approaches also provide a lossless mode, but still use computationally expensive transforms, which have a processor load that is too high for storing the data stream from the telescope. Three purely lossless compression methods exist: Véran and Wright [11] proposed an approach that accounts for the noise in astronomical images as it separates high-order bits from noisy - and by implication, less compressible - low order bits. Sabbey describes an adaptation of the RICE compression method [8] using linear prediction, for compressing astronomical images [9]. Finally, Weghorn et al. [12] describe a method for applying lossless compression in astronomy to interferometric data using a scheme called Signed Huffman Coding. The commonly used image format in astronomy - the FITS (Flexible Image Transport System) format [2] - currently includes only limited support for size compaction techniques. The work described here aims to develop a lossless and computationally efficient compression method by segmenting the image into regions, which are compressed adaptively by different methods. As a result, the overall-size of the stored image shall be minimized.

3 Compression by separation and region-adaptive bit allocation

Lossless compression techniques attempt to identify and exploit properties of the data for compaction [10]. The properties are described using a model of the data. One straightforward model is presented here to show the possible compression by describing the data using this model. The image-signal i(x) of an astronomical image, in general consists of three additive, independent parts: the signal s(x), which is generated by photons absorbed and detected by the light sensor, electronic or thermal noise n(x), and a constant signal offset value b, which is called bias.

$$i(x)=s(x)+n(x)+ b. \tag{1}$$

The positive bias is required to avoid negative output values in the dark areas of the image, as the noise includes negative values. Ideally, the signal generated by the incident light s(x), should be large compared to the noise, for obtaining a good signal-to-noise ratio (SNR). The noise in professional astronomical photography typically influences fewer than the lower 4 bits of a 20-bit value generated by the digitizer of the imaging device. Hence, depending on the signal s(x), two image areas can be distinguished: regions of signal (ROS), where s(x) is significantly larger than n(x), and background regions, where s(x) carries values close to zero.

Pixel values along a sample line section (grey line in Fig. 1(a)) through the image are shown in Fig. 1(b), where the constant bias and the dynamic range of the noise portion are marked. Fig. 1(b) clearly indicates that the full dynamic range, provided by the FITS file format, is only required within the bright areas of the image where the signal component s(x) is large. Separating the background areas from ROS, and coding both parts differently leads to a significant reduction of the storage space. In typical astronomical photographs, the dark area, which can be coded with very few bits, fills about 90 percent of the image. A sample calculation, using the raw data presented in Fig. 1(a) shows that with a region-adaptive bit allocation, a significant reduction of the required storage space is achievable. ROS and background regions have to be identified by analysing the image signal shown in Fig. 1(b); the bias of this image is 24, while the noise portion has a dynamic range smaller than ± 8 intensity steps. Therefore, the background noise can be coded with 3 + 1 bits of the original 32 bits. Regions of signal have to be coded by the full dynamic range of 32 bits (second term in formula (2)). 79.2 percent of the image corresponds to background while 20.8 percent form the ROS. Even if a bitmap - requiring 1 bit of the original 32 bits/pixel for the whole image (third term in formula (2)) - is used to describe the different image regions, the size compaction that can be achieved in this example is nearly 3:1.

$$0.792 \cdot \left(\frac{3+1}{32} \right) + 0.208 \cdot \left(\frac{32}{32} \right) + \left(\frac{1}{32} \right) = 0.338 \tag{2}$$

The compression ratio estimated in formula (2) is achieved even without using well known compression methods. Applying compression to both regions will lead to further size reduction. Another possibility for enhancing the compression is to use more efficient methods to describe the different image regions. Such methods [4, 14, 15] could further reduce the image size by reducing the last term in formula (2). Hence, a range of standard and astronomy specific techniques, as well as techniques from second generation image compression will be examined to efficiently code the data of both areas. Therefore this method can be considered to be a version of second generation image coding.

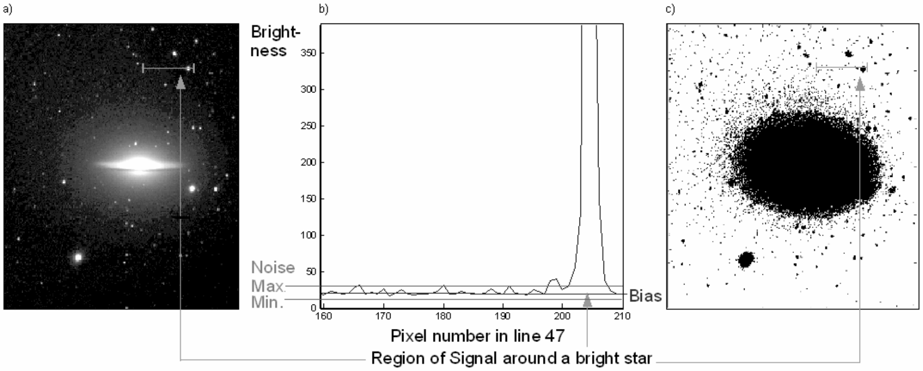


Fig. 1: a) Astronomical photograph sample. b) Signal plot of an image scan line (denoted by the grey line within the photograph given in a). c) The decision map for background noise and signal pixels.

4 Conclusion and further work

The sample calculation presented here, shows that a significant compaction can be achieved by segmentation and region-adaptive bit allocation. Future work will focus on the development of a computationally efficient compression algorithm, which determines the image bias, the regions of signal and the background area, and encodes these different parts separately. To develop a compression technique that takes advantage of image properties, first the determination of the image bias has to be automated. This work appears complicated by the fact that the bias in general has a tiled and non-uniform shape. Additionally, a range of image peculiarities have to be considered. Hence, in the first working steps, an automated compensation method for the bias will be developed before the core work on efficient compression can start.

References

1. S.G. Djorgovski, R.J. Brunner, A.A. Mahabal, S.C. Odewahn, R.R. de Carvalho, R.R. Gal, P. Stolorz, R. Granat, D. Curkendall, J. Jacob, and S. Castro, Exploration of large digital sky surveys, Mining the sky: Proceedings of the MPA/ESO/MPE Workshop, Garching, 31 July - 4 August 2000 (Heidelberg), ESO Astrophysics symposia, vol. 6, Springer, 6 2003, pp. 305 - 322.
2. W. D. Pence, FITS homepage, September 2004, http://fits.gsfc.nasa.gov/ (28.02.2005).
3. Steve B. Howell, Handbook of CCD astronomy (Cambridge observing handbooks for research astronomers), Cambridge University Press, Cambridge (UK), 2000.
4. D. Huffmann, A method for the construction of minimum redundancy codes, Proceedings of the IRE 40 (1952), no. 9, 1098–1101.
5. F. Murtagh, J. Starck, and A. Bijaoui, Multiresolution in astronomical image processing: a general framework, International Journal of Imaging Systems and Technology (1995), 332 - 338.
6. W. H. Press, Wavelet based compression software for FITS images, ASP Conf. Ser. Astronomical Data Analysis Software and Systems I (D. M., Worrall, C. Biemesderfer, and J. Barnes, eds.), vol. 25, Astronomical Society of the Pacific, January 1992, p. 3.
7. P. J. Quinn and K.M. Gorski (eds.), Toward an international virtual observatory, Springer, Berlin; Heidelberg, 2004.
8. R. F. Rice and J. R. Plaunt, Adaptive variable-length coding for efficient compression of spacecraft television data, IEEE Transactions on Communication Technology 6 (1971), no. 19, 889 - 897.
9. C. N. Sabbey, Adaptive, lossless compression of 2-D astronomical images, ASP Conference Series (San Francisco) (D. M. Mehringer, ed.), vol. 172, Astronomical Society of the Pacific, 1999, p. 129.
10. David Salomon, Data compression: the complete reference, 2 ed., Springer, New York; Berlin; Heidelberg, 2000.
11. J. P. Véran and J. R. Wright, Compression software for astronomical images, Astronomical Data Analysis Software and Systems III ASP Conference Series (D.R. Crabtree, R.J. Hanisch, and J. Barnes, eds.), vol. 61, 1994, p. 519.
12. H. Weghorn, J. Colberg, and T. Bauer, Real-time image compression method for astronomical images, Science with the VLT Interferometer ESO Proceedings (F. Paresce, ed.), Springer-Verlag, Berlin Heidelberg, 1996, pp. 399 - 400.
13. R. L. White, High-performance compression of astronomical images, Proceedings of the NASA Space and Earth Science Data Compression Workshop (Snowbird (Utah)) (James C. Tilton, ed.), March 1992, p. 403.
14. J. Ziv and A. Lempel, A universal algorithm for sequential data compression, IEEE Transactions on Information Theory 23 (1977), no. 3, 337 - 343.
15. J. Ziv and A. Lempel, Compression of individual sequences via variable-rate coding, IEEE Transactions on Information Theory 24 (1978), no. 5, 530 - 536.

Efficient Digital Image Restoration for Analog Video Tape Archives

Thilo Bauer

Bluewater Multimedia Concepts,
Kölnstraße 191, D-53757 St. Augustin, Germany
tbauer@bwmc.net

Abstract. A detailed research of the image restoration process for digitization and archiving of analog video tapes will be presented. A new complete model of image restoration will be developed and applied to real video sequences. The research will focus primarily upon the digital imaging and restoration process chain based on real hardware issues. Our work also includes a refinement of common image restoration techniques. Preliminary results of our investigation already show detailed aspects on how to set up a new restoration process and opportunities to define ranking of issues to be solved within this framework.

1 Introduction

There is still a considerable amount of movies produced with analog video tapes by contemporary artists, public organizations, science and individuals. The discussion about whether and how to preserve, restore and store the contents digitally is still in progress. A common consent can be found as a preliminary result, which is very important to define the next steps [2]. The original video tape content is expected to decrease in quality by time and caused by material properties. The most important issue is to preserve the movie content as an intellectual property and to restore contents which suffer from system intrinsic and aging related image degradation. It is of less importance to preserve the tape material itself. Preliminary results of our own work confirm these statements about a general restoration framework.

2 About this project

For improving the digital archive process we primarily focus on the image restoration process for analog video tapes. The topics of this project are:

- A description of the main effects of image degradation
- Questioning: collect users experience about symptoms of image degradation
- Ranking of the effects to be compensated
- Development of a refined theoretical model of video image degradation
- Field tests: application of the software to different videos and evaluation by a group of viewers
- Implementation of related image restoration algorithms

3 Refinement of digital image processing of analog video

A theoretical model of the image degradation shall be developed. The model will also introduce a refinement of common removal techniques for noise, additional side effects and terms which can be detected and separated from the video image content. Based on this theoretical model we will develop and implement new algorithms and removal techniques to form a complete process. The algorithms will then be applied to collected tape material from different sources and evaluated by a group of users.

In a first analysis, the following effects of image degradation have been identified by ourselves: Coloring (color mismatch), color out-bleeding, resistor and tape noise, color speckle noise, drop-outs (scratches), oversampling or undersampling of the signal, image blur, oscillations, image flicker, sampling noise and errors, additive signals from environment and/or the apparatus itself (e.g. periodic interference), video line jitter, total loss of image synchronization. It is important to notice, that current video image processing techniques do not take real hardware into account. For example noise suppression techniques assume a white Gaussian noise. Therefore, some of these effects mentioned above are not corrected with these image restoration techniques. Under certain circumstances the algorithms will destroy content structures, e.g. depending on the threshold for a simple noise removal algorithm. New image correction algorithms will be developed based on our theoretical model. These methods will be tested within simulated video sequences and also applied to restore real video tape recordings.

4 Image defects and restoration techniques

The Wavelet transform is a state-of-the-art tool for noise removal [3, 7, 11]. There remain other noise terms, like color speckle noise. Color speckle noise appears as a smooth variation of the color or grayscale intensity. The effect is assumed to be a result of a sum of resistor and tape noise, additional periodic interference, and oscillations at different frequency scales.

Coloring means to harmonize different tape material within the production process [4]. This includes white balancing, artificial coloring, and also color matching when editing and pasting scenes of a movie from different sources.

Fourier methods are used to detect and remove moiré at the intensity level of the content [9, 10]. However, in a noise analysis, the intensity of these frequencies is very low compared to the contribution of the image content. One solution of detecting these frequencies would be to analyze black movie frames. However, it appears unrealistic to find such frames in each video content. We propose combining Fourier and wavelet methods to analyze the deviations of the image signal to collect information about periodic terms from content frames.

Intensity variations of the analog video signal will introduce oscillations caused by electronics, e.g. dark shadows of bright image structures in direction of the video scan line. These oscillations shall be described by a physical model.

Spikes and drop-outs (scratches) require a different and destructive approach of error removal. Abbas and Domanski define a very simple synthetic model of spike appearance applied with their removal process [1]. As mentioned above spikes and drop-outs will also introduce oscillations, as they introduce large variations of pixel intensity. Having measured the oscillations the drop-out removal is expected to obtain better results.

Video image flicker will show up as a variation of the bias of consecutive frames and video scan lines. The effect of video image flicker is different from the flicker found with film movies [6]. The time-scale of the variations is smaller than the frame rate. Thus, these variations will be detected within a single frame.

Video line jitter is a variation of the video image and line synchronization (image stabilization). Worst case is an image roll-through, if the hardware is unable to restore the image sync signal. Best (usual) case is a horizontal misalignment of consecutive video lines by a few pixels or within sub-pixel range. Current research in removal is ongoing and does not seem to be understood completely. The results from a Bayesian approach [8] and earlier work [5] yielded contradictory results and left open questions. Professional digitizers use hardware implemented time-base correctors (TBC). A TBC is a threshold detector which restores the line sync signal. Thus the success of a sync restoration still depends on side effects of the signal variation. On the other hand, it is impossible to access these regions of the analog video signal with factory built video digitizer hardware. Therefore, it is proposed to perform a detailed analysis to clarify whether there are correlations between video line jitter and additional terms described so far, which can be identified within the digitized content sub frame.

There remain some effects, like color out-bleeding. This effect was mentioned by users and appears as a blur of the blue or red colors. the effect shall be analyzed regarding its source and constraints. People suggest that the effect relies on tape machine properties and will also depend on different TV formats, like PAL or NTSC. On the other hand, we know similar effects caused by the camera optics of CCD video camcorders (i.e. defocused image within the near infrared).

5 First results

For defining and implementing the complete image restoration process it is important to find out, which of these effects will be noticed and classified by users, which are divided into two groups: producer (e.g. TV production, artists) and consumer (e.g. people watching TV).

As a preliminary result we collected new data from questioning TV journalists. Asking them about video tape restoration, people suggested that this will mean a mechanical tape restoration. This is a result of (a) the traditional manual work with film material in the past decades, and (b) trying to do some similar approach with destroyed tapes. The common process of archiving video tape material is to achieve tape copies of existing material on a regular time base (tape copies either analog-analog or analog-digital). People mention, that tape material will likely have failure and material defects and will be heavily influenced by aging even within a short period of time. They also mention the ongoing degradation of quality with each new tape copy. The worst case is a complete destruction, caused by abrasion of the magnetic tape, even with new tapes. It is interesting to notice, that the issues discussed within this paper are not only related to historic materials, but also on modern video tapes. Asking about image degradation of analog video tapes and how to proceed with these videos, people will become curious and mention at least two (up to six) effects listed above to be corrected. The most common effects mentioned are scratches and noise. It is interesting to see, that different appearance of noise, like color speckle noise, are not differentiated, although they can easily be noticed. Other effects, like a complete image roll-through (loss of sync), are well known, but do not seem to gain focus, if asking about priority. Until now, it is not quite clear, why these effects do not gain focus of interest, or what really happens with these tapes. From private communication with people, we suggest that users easily notice this image defect, and therefore do not use these videos tapes for further production. The quality of tapes suffering from this effect is said being absolutely unacceptable.

6 Conclusion

Important effects of image degradation of analog video tapes are described. Possible solutions are proposed within this paper. A questioning of users confirmed well known issues to be solved within the context of the digital archive process. Our preliminary results already show a possible ranking of these issues. It is proposed to extend the questioning to different user groups, like video artists and end-user. This will lead to a final theoretical model and a concept for the digital image restoration process which then shall be implemented and tested.

References

1. Jamal K. Abbas and Marek Domanski. Rejection of Scratches from Video by use of Filters with Prediction Error Processing. Proc. Of Signal Processing X: Theorie and Applications, Eusipco, 2000.
2. Erich Gantzert-Castrillo. The Archive of Techniques and Working materials Used by Contemporary Artists. WAAC Newsletter, 21, 5 1999.
3. Aglika Gyaourova, Chandrika Kamath, and Imola K. Fodor. Undecimated wavelet transforms for image de-noising. Technical report, Department of Computer Science, University of Nevada, Reno, 11 2002.
4. Jianying Hu and A. Mojsilovic. Optimal color composition matching of images. Proceedings 15th International Conference on Pattern Recognition, 2002.
5. Gang Luo, Opas Chutatape, and Hui Fang. Experimental study on nonuniformity of line jitter in CCD images. Applied Optics, 40(26):10, 9 2001.
6. Francois Pitié. Removing Flicker from Old Movies, 10 2002.
7. Aleksandra Pizurica, Vladimir Zlokolica, and Wilfried Philips. Noise Reduction in Video Sequences Using Wavelet-Domain and Temporal Filtering. SPIE Proc., 2003.
8. Jianhong Sheng. Bayesian Video Dejittering by BV Image Model. CAM Report, 02:64, 2002.
9. Denis N. Sidorov and Anil C. Kokaram. Suppression of Moiré Patterns via Spectral Analysis. SPIE, 2001.
10. Denis N. Sidorov and Anil C. Kokaram. Removing Moiré from Degraded Video Archives. Eusipco, 2002.
11. Ivan W. Selesnick and Ke Yong Li. Video denoising using 2D and 3D dual-tree complex wavelet transforms. SPIE Proc., 2003.

Speech Recognition, Speaker Recognition, and Speech Synthesis - A Unified Technology Approach

Dirk Saller

Berufsakademie Mannheim, Informationstechnik
Coblitzweg 7, 68163 Mannheim, Germany
saller@ba-mannheim.de

Abstract. Speech signals turn out to have in most parts an almost periodic signal structure. There is evidence for a few single periods of a speaker (having a length between about 1/1000 to 1/100 of a second) to encode enough information to (i) recognize the information, (ii) recognize the speaker and (iii) reproduce the speakers voice, partially. The goal of this paper is to show that, within this approach, only a few data is needed to predict, with high probability, spoken information (speech-to-text) and speaker's identity (speech-to-speaker).

1 Introduction

Looking at speech signals in a high resolution, one can see that about 60-80% of the speech signals have an almost periodic structure coming from the oscillation of vocal cords. Analyzing the elements of this almost periodic structure it turns out that such periodic pieces carry some characteristic information on the speaker and on the information. From this fact several questions arise:

(i) can we recover from one periodic piece the spoken text?
(ii) can we recover from one periodic piece the speaker's identity?
(iii) If (i) and/or (ii) is true, can we raise the probability of the detection method by combining different periodic pieces for the same speaker or the same information, respectively?

The answer to theses questions can be found by analyzing step by step the role of the (almost) periodic pieces within a speech signal.

2 Speech signals at a high time resolution

In the graphic function plots Fig. 1 to Fig. 3 of an audio signal for the text "Willkommen in Stuttgart" (= "welcome to Stuttgart" in German) spoken by a female's voice, stepwise is zoomed in to the final resolution of 1/1000 s. In the last audio signal graph (Fig. 3), showing the speech signal for the part "il" out of "Willkommen in Stuttgart", an almost periodic behavior is found.

In particular, there is a continuous transition of one period (basic element) to another period, belonging to the information "i" and "l", respectively. The basic elements in the middle look like a nonlinear superposition of the boundary elements.

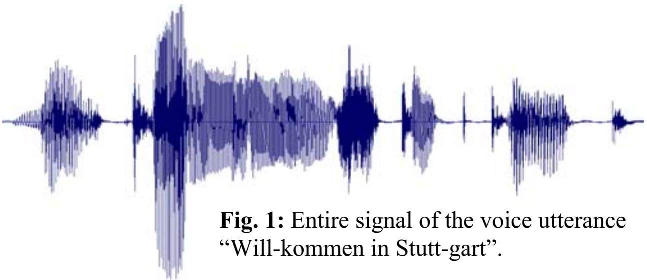


Fig. 1: Entire signal of the voice utterance "Will-kommen in Stutt-gart".

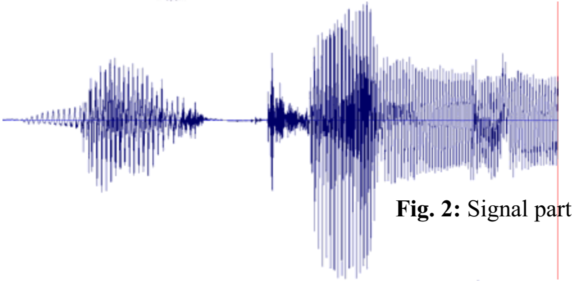


Fig. 2: Signal part of "Will-kommen".

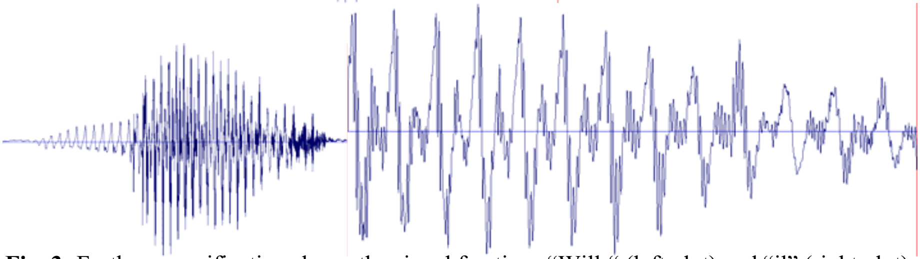


Fig. 3: Further magnification shows the signal fractions "Will-" (left plot) and "il" (right plot), in which a periodic signal structure clearly can be identified.

It is supposed that only a few basic elements of a speaker are needed to recover with a high probability the (quasi periodic part of the) information of a spoken text (of this speaker) and his/her identity.

3 Components for constructing an application

In order to investigate in a professional and efficient manner the above hypothesis, some automation of processes is necessary. In particular, three components turn out to play a crucial role in all of the three technologies that come out from the signals' characteristic structure, namely the speech and speaker recognition and the speech synthesis.

The first component consists of a "periodicity finder", which extracts for a given trigger, almost periodic elements from an arbitrary speech signal, cutting and extracting them as elementary pieces (wave files with only one or a few almost periodic pieces).

The second component consists of the "feature extractor". Its role is to generate some characteristic numbers for any basic element. It makes the basic elements accessible to computations and comparisons.

The third component is a toolbox for teaching the systems in classifications for a given set of “similar” basic elements. Here, the term “similar” is defined by the trainer of the classifying model. In particular, a teacher of a classifier tells the classifier, which basic elements are considered to be in the same equivalence class. Roughly speaking, the teacher defines an equivalence relation.

4 Some results of classifications

Table 1 contains the results of several classification scenarios. It turns out that these basic elements are good candidates for containing enough characteristic data of speaker and information.

Input class of phonemes	Classification rate
personA [phA,phE,phU,phO] vs. personB [phA,phE,phU,phO]	99.4048 %
personA [phA] vs. personB [phA]	97.619 %
personA [phE] vs. personB [phE]	100 %
personA [phO] vs. personB [phO]	100 %
personA [phU] vs. personB [phU]	97.561 %
personA [phA] vs. personA [not phA]	85.5072 %
personB [phA] vs. personB [not phA]	93.9394 %
personA [phA] vs. [phE] vs [phO] vs [phU]	85.5072 %
personB [phA] vs. [phE] vs [phO] vs [phU]	83.8384 %
personA,personB [phA] vs. [phE] vs [phO] vs [phU]	79.7101 %

Table 1: Measurement of recognition rates for different scenarios.

5 Next steps

The speech data base will be expanded to more than just two persons. In particular, a significant quantity of speakers and information will be considered to get a prototype for the applications. Combinations of different classification models will be investigated to raise the detecting probability.

Section on Miscellaneous Subjects

A Java Class Library for Stepwise Execution and Visualization of Algorithms

Olaf Herden

University of Cooperative Education Stuttgart, Campus Horb
Department of Information Technology
Florianstr. 15, D-72160 Horb/Neckar
o.herden@ba-horb.de

Abstract. During basic courses in algorithms and data structures many algorithms have to be tought by lecturers and learned by students. The classical way is describing the algorithms in pseudocode and showing examples with chalk on the blackbord or with some animated slides with presentation programs. More appropriate is learning the algorithms by an interactive software component where the algorithm is executed stepwise and additionally visualized. In this paper we present an approach to realize the latter way.

1 Introduction

Many algorithms, e.g. sorting, searching and basic graph algorithms have to be tought from lecturers to students in basic courses. The typical way of imparting this knowldege is describing the algorithms in pseudocode and showing examples with chalk on the blackbord or with some animated slides with presentation programs like Microsoft PowerPoint. For repetition and comprehension the students have to create their own examples and "execute" them with pencil and paper.

A more appropriate way is learning the algorithms by using an interactive software component where the algorithm can be executed stepwise and is ideally visualized. The stepwise execution ideally should be able forwards as well as backwards. Moreover, the parametrization of the algorithm would be beneficial.

A second motivation for this project is finding an interesting task for student projects for the programming lecture which is typically taught in parallel to the algorithm and data structures course.

A third aspect is use of new media techniques and e–learning concepts in education as additional aspect to conventional lectures in the classroom.

In this paper we present an approach where important steps of algorithms are written to a log structure and afterwards this log can be navigated for stepwise execution of the algorithm.

2 Concept

This section describes the main concepts. While in 2.1 the Java class library is sketched section 2.2 describes the steps of the process of applying this library. Finally, section 2.3 demonstrates the result of this application by an example.

2.1 Class Library

The class library is depicted in figure 1.

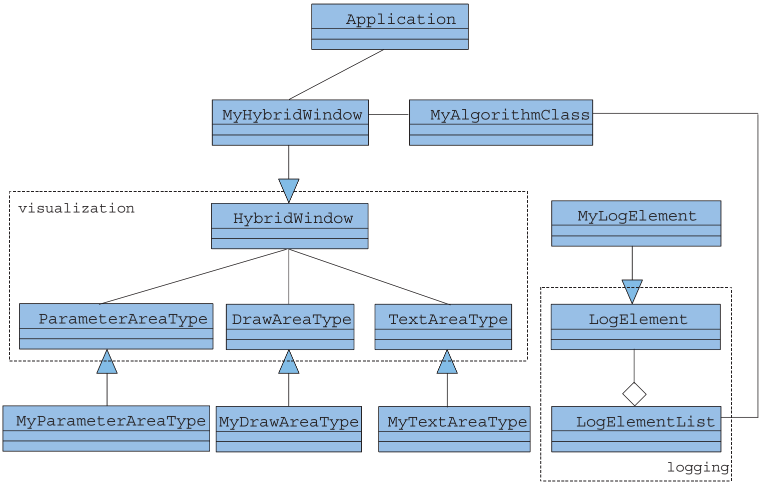


Fig. 1. Class Library

The library consists of the two packages *visualization* and *logging*. Package *logging* provides a class *LogElement*. An instance of this class consists of the relevant information of an algorithms single step. *LogElementList* is a collection of *LogElement* objects.

In package *visualization* a generic window for the output is realized. This window is called hybrid window because it can be started as desktop application as well as applet in a browser. Among other things a *HybridWindow* object contains a parameter area (class *ParameterAreaType* for user input of probably existing parameters for the algorithm), a text area (class *TextAreaType* for textual output during the stepwise execution) and a draw area (class *DrawAreaType* for graphical output during the stepwise excecution). An instance of *HybridWindow* provides selecting parameters, executing the algorithm and navigating forwards and backwards through the log element list and producing textual and graphical output during this navigation.

All classes outside these two packages are individual to the concret application and will be described in the following subsection.

2.2 Process of Application

In this section the process of applying the packages introduced in the former section to build an application is described step by step.

1. Implement your algorithm
   First you have to implement the algorithm that should be executed stepwise. In figure 1 this should be done in the class *MyAlgorithmClass*.
2. Extend class *LogElement*
   By creating the class *myLogElement* the basic LogElement which only contains a step number and a string for the description can be enriched by all the information being relevant for your algorithm.
3. Implement a logged version of your algorithm
   Create a version of the algorithm from step 1 where all the relevant information are logged, i.e. they will be written to an object of *LogElementList*.
4. Extend the GUI elements
   Extend the four GUI elements *ParameterAreaType*, *DrawAreaType*, *TextAreaType* and *HybridWindow* by all the information that should be displayed during the stepwise execution in the window. This extension is sketched in figure 1 by the versions of the classes with the prefix *My*.
5. Build your application
   Finally, create a class *Application* where an instance of class *MyHybridWindow* is created and started to carry out the stepwise execution and visualization.

2.3 Example

Figure 2 shows an example [2] of applying the process described above. The realized algorithms are graph coloring algorithms. Left hand side we can see the parameter area where the kind of the algorithm (greedy coloring or backtracking) can be selected. In the center of the window we can see the visualization of the colored graph and on the right the textual output. The buttons in the bottom area offer navigation for stepwise execution of the algorithm.

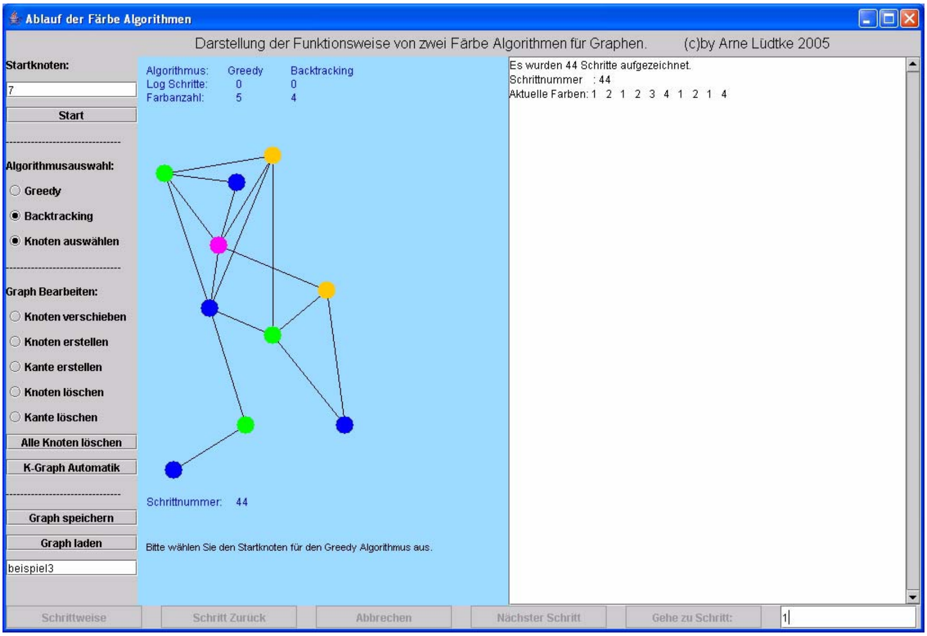
3 Summary and Outlook

Up to now some sorting and graph algorithms have been realized in student projects. This was an adequate task for the students and the results can be used for demonstrating these algorithms in following courses. So the students can comprehend the algorithms on their own and use the components for preparing for written exams.

The project will be continued. As future tasks we see the implementation of more algorithms like those described in [3][1][4] and the integration of the solutions realized so far into an overall e–learning concept for supporting education in algorithms and data structures.

References

1. Knuth, Donald: *The Art of Computer Programming, Vol. 3.* Addison–Wesley, 2nd edition, 1998.



Ablauf der Färbe Algorithmen

Darstellung der Funktionsweise von zwei Färbe Algorithmen für Graphen. (c)by Arne Lüdtke 2005

Startknoten: 7

Algorithmusauswahl:

○ Greedy

● Backtracking

● Knoten auswählen

Graph Bearbeiten:

○ Knoten verschieben

○ Knoten erstellen

○ Kante erstellen

○ Knoten löschen

○ Kante löschen

beispiel3

Algorithmus:	Greedy	Backtracking
Log Schritte:	0	0
Farbanzahl:	5	4

Schrittnummer: 44

Bitte wählen Sie den Startknoten für den Greedy Algorithmus aus.

```
Es wurden 44 Schritte aufgezeichnet.
Schrittnummer   : 44
Aktuelle Farben: 1  2  1  2  3  4  1  2  1  4
```

Gehe zu Schritt: 1

Fig. 2. Example of a window for graph coloring algorithms

2. Lüdtke, Arne: *Graph Coloring Algorithms.* Project Report, University of Cooperative Education Stuttgart, Campus Horb, Department of Information Technology, 2005.
3. Sedgewick, Robert: *Algorithms in Java, Vol. 1+2.* Addison–Wesley, 2003.
4. McHugh, James A.: *Algorithmic Graph Theory.* Prentice Hall, 1989.

Approaches to Vision Based Emotion Recognition Using Neural Networks

Stefanie Maoro, Dirk M. Reichardt

Berufsakademie Stuttgart, University of Cooperative Education, Information Technology,
P.O.Box 100563,D-70004 Stuttgart
reichardt@ba-stuttgart.de
http://www.ba-stuttgart.de/dr

1 Introduction

In human communication the nonverbal part is more essential for a correct understanding of the transmitted message. A key component of the message of a sender can be seen in the facial expressions shown. One component for interpretation is the emotional state of a person which we try to conclude from facial expressions. Basic emotions are reflected reliably by facial expressions [1]. The basic emotions we consider in our work are anger, disgust, fear, joy, sadness and surprise. In a previous publication [2] we discussed a small research framework for emotion recognition from facial expressions at the Berufsakademie Stuttgart, University of Cooperative Education, and first results using the Dempster Shafer rule [3]. In this paper we present first results using a neural network for the classification of basic emotions from preprocessed images.

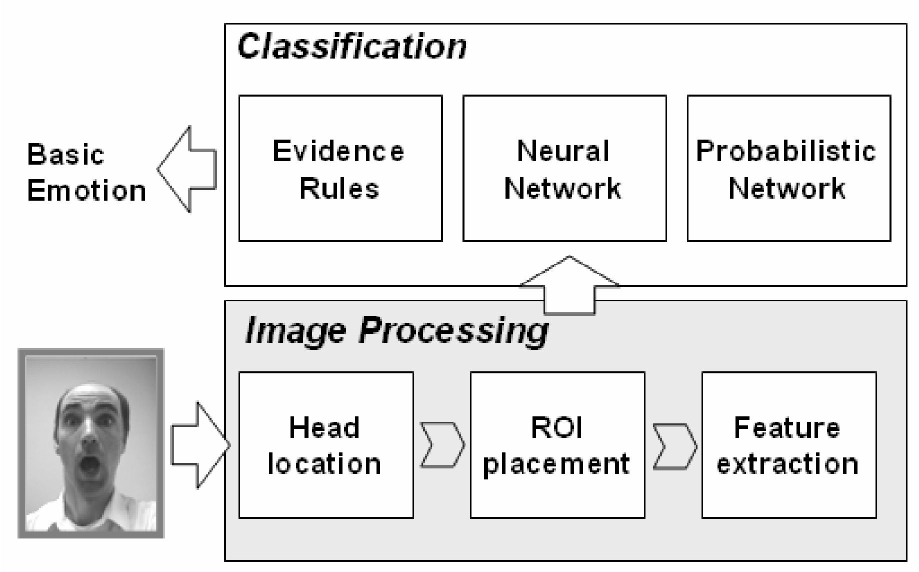


Fig. 3: Research framework for emotion recognition from facial expressions.

2 The Image Recognition Framework

Our system integrates image processing methods and AI methods in a common framework. The design comprises: head location, identification of regions of interest (ROI), feature extraction, and emotion classification. The neural network approach presented here uses the results provided by a set of algorithms for dedicated image region analysis [4]. Unlike the previous approach based on the Dempster-Shafer rule, the neural network uses continuous instead of discrete results of the image processing phase. Nevertheless, both will be compared and combined in the overall system starting from a set of test images. Since the framework serves as an educational experimental tool as well, it includes probabilistic networks as a third classification component which will however not be considered in this paper.

3 Specification and Network Architecture

The network we use is a three layered back propagation network. The network input is provided by the image processing modules. The following 12 input variables are used: LEO/REO (left/right eye opening), LDB/RDB (left/right eye distance to brows), HDE (height difference of the eyes), FW (forehead wrinkles), NWH/NWV (nose wrinkles horizontal/vertical), CWL/CWR (cheek wrinkles left/right), MO (mouth opening), and TV (teeth visible).

Therefore the network contains twelve input layer neurons. First experiments showed that the (typical) number of 25 hidden layer neurons is sufficient1. As for the output layer we provide a neuron for each of the basic emotions anger, disgust, fear, joy, sadness and surprise plus a neuron for the often used emotion ‘disdain’.

4 Training and First Classification Results

First tests have been carried out only on a small set of 80 labeled images covering the basic emotions mentioned above. After some variations of learning rate and thresholds the following first results could be computed, which are comparable to those achieved by the Dempster-Shafer approach. The table below shows a typical classification result of the network. The verification value was obtained by presenting the images to a group of students. It provides the percentage of correct classifications by humans which is important for determining the quality of the algorithms.

The comparison with the “human” classifier shows good results already. Typical mistakes (like confusing fear and surprise) often appear in human interpretation. Nevertheless, more labeled images for test and training are required in order to come to more significant results.

[1] Even lower numbers of hidden layer neurons provided comparable results on the (small) given set of test images.

Emotion	**Neural Network**		**Verification**	
(Image)	**1**	**2**	**1**	**2**
Anger	Anger (0.18)	Disdain (0.06)	Anger (0.63)	Disdain (0.18)
Disgust	Disgust (1.00)	-	Disdain (0.48)	Disgust (0.30)
Fear	Fear (0.93)	Surprise (0.02)	Surprise (0.96)	Fear (0.04)
Joy	Joy (0.97)	-	Joy (1.00)	-
Sadness	Sadness (0.45)	Disgust (0.13)	Disdain (0.52)	Disgust (0.30)
Surprise	Fear (1.00)	Joy (0.02)	Surprise (0.83)	Joy (0.17)
Disdain	Fear (0.95)	Surprise (0.33)	Disdain (0.66)	Disgust (0.13)

Fig. 2: Comparison of neural network and verification test set with "human" classifiers. Verification numbers stand for percentage, network numbers stand for the associated output neuron (range [0,1]).

5 Conclusion

Due to the low number of images in training, the network results cannot be interpreted as generally veritable but give a first impression how close to emotion recognition we can get by using a standard neural network approach.

Future work, supported by student research projects, will concentrate on the improvement of the first phases, enhancing the set of features extracted by the image processing modules, and improvement of the algorithms used. Furthermore, an integration of facial feature recognition with ongoing research on gesture analysis and speech analysis will be our next goal.

References

1. Andrew Ortony, T.J. Turner: What's basic about basic emotions? Psychological Review, 1997, p. 315-331
2. Dirk M. Reichardt: "Applying Computer Science Methods to User Emotion Recognition", in: Weghorn (ed.), Proceedings of the 1st Annual Meeting for Information Technology & Computer Science at the BA-University of Cooperative Education, Stuttgart, Germany, 2004, ISSN 1614-2519
3. Glenn Shafer: A Mathematical Theory of Evidence, Princeton University Press, 1976
4. Martin Joos, Philipp Krauss: Computergestützte Mimikerkennung mit fortgeschrittenen Bilderkennungs- und Bildverarbeitungsalgorithmen unter Java, Studienarbeit, Berufsakademie Stuttgart, 2004

Object–Oriented Tokens for the Parser Generator jay

Holger Haas, and Martin Plümicke

University of Cooperative Education Stuttgart
Department of Information Technology
Florianstraße 15, D–72160 Horb
m.pluemicke@ba-horb.de

Abstract. jay is a parser generator for JAVA derived from ATTs' parser generator **yacc**, which was designed for the C programming language. Being based on **yacc**, jay is not well adapted for object-orientation, because this principle is not known in the C-world. To correct this flaw jay was redesigned to cope with objects.

1 The Parser Generator jay

The parser generator jay [1] is a tool which generates for a given grammar in BNF a respective parser in JAVA. The specification of jay is very similar to that of yacc [2]. This means that it is very easy to learn jay for a yacc user. But there are some significant differences, which arises from the fact that JAVA is object–oriented and the original yacc bases on C:
The generated JAVA program is parameterized by the JAVA interface `yyInput`:

```
boolean advance () throws java.io.IOException;
int token ();
Object value ();
```

The method `advance()` reads the next lexem and gives false if end-of-file is reached. The method `token()` gives the corresponding token. Finally, the method `value()` is used to get the corresponding object for the jay's value–stack. This interface must be implemented by the user, himself.
For the implementation of `advance()` and `token()` a scanner is needed. The simplest possibility is to implement a subclass of the JAVA's stringtokenizer. More comfortable is for example to use JLex [3]. JLex is a generator for a lexical analyzer. It bases on the well-known lex [2] but produces JAVA code and is written in JAVA. The design approach of JLex is object–oriented, which means that the produced tokens can be real objects (instances of classes, not only `int`s). In contrast jay accepts, as yacc, only int values as tokens.

2 The New Design

The original design of jay has the disadvantage, that attributes of Tokens, which are recognized by the scanner (e.g. the name of the identifier in the token identifier), must be handled by the user, himself. These attributes must be treated

by the implementation of the interface `yyInput`. Moreover, the whole handling of the interface `yyInput` is inconvenient and can be automated.
Our approach changes the type of the tokens in jay from the base type `int` to a subtype of a new type `yyTokenclass`. This means that for each declaration `a subclass tok_name of yyTokenclass` is generated. The class `yyTokenclass` contains a field `value` of the type `Object` and a constructor to assign values to this field. This means that a typical statement in the JLex specification is `return tok_name(attribute)`, where `attribute` contains the token's attribute, as e.g. the name of the scanned identifier.
An overview of the generated class structure is presented in figure 1. The JLex

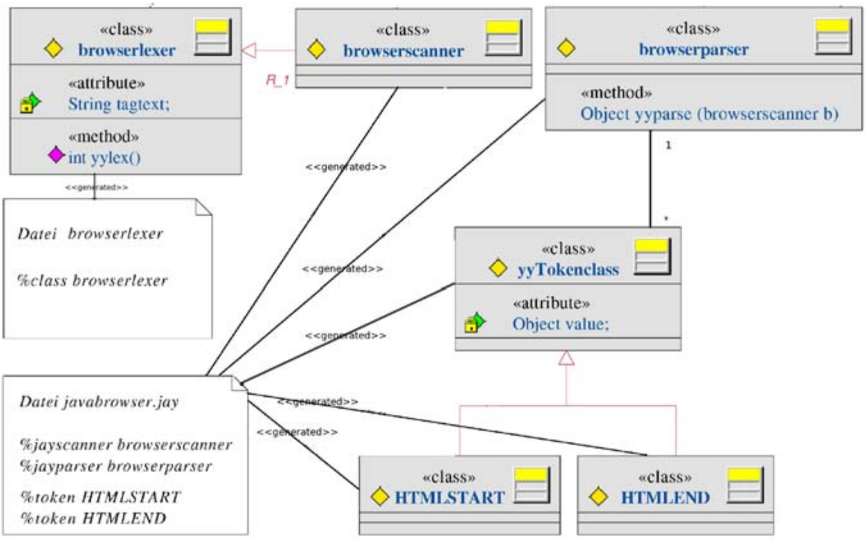


Fig. 1. The generated class structure of new jay and new JLex

specification `browserlexer` generates the class `browserlexer`, which contains the scanner method. The directive `%class` defines the name of the generated class. The other classes are generated by the new jay. The new directive `%jayscanner` produce a class, which represents the scanner as a subclass of the class generated by the JLex specification. The additional new directive `%jayparser` produces the parser class with the method `yyparse()`. Finally, for each declared token a class is generated as a subclass of the also generated class `yyTokenclass`.

3 The Implementation

With the new design, two goals were to be achieved:

1. The tokens should be handed over from the scanner to the parser as objects
2. The user should not longer need to implement the scanner-interface himself

To make this possible, the source code of jay and JLex had to be modified in several ways:

skeleton jay uses a file named ”’skeleton”’ as template for the interpreter code. It contains code fragments which determine among other things the token types, token handling and the scanner-interface simply by copying the corresponding parts to the generated parser code in the progress of the parser generation.
In the modified skeleton the scanner-interface does not longer contain the methods `int token ()` and `Object value ()`. With the use of object tokens, the whole parsing can be handled with the method `yyTokenclass advance ()` alone. If end-of-file is reached, `advance ()` returns `NULL`.
The superclass `yyTokenclass`, from which the token objects inherit their attributes, is also defined in the new skeleton.
The generated cellar automaton works internally still with integer values. The interpreter code had therefore to be adjusted to allow the utilisation of token objects.

jay The source code of jay had to be modified in a way that it does not longer create integer constants as tokens, but token objects which contain the corresponding integer constant as attribute. This attribute can then be processed by the cellar automaton as supplied before.

JLex In order that the user does not longer have to implement the scanner-interface himself, the source code of JLex had also to be adjusted. The modifications affect mainly the ability to process the new directives `%jayparser` and `%jayscanner`. If these directives are found, the new JLex implements the scanner interface. If not, JLex behaves as the original version.

4 Summary and Outlook

With the mentioned modifications jay and JLex can be used and understood more easy and, which is more important, fit into an object–orientated concept. However, the concept of encapsulation was not slightly neglected. For further development, improvements of encapsulation should be a challenge.

References

1. Kühl, B., Schreiner, A.T.: jay – Compiler bauen mit Yacc und Java. iX (1999) (in german).
2. von Levine, J.R., Mason, T., Brown, D.: Lex & yacc, UNIX Programming Tools. A Nutshell-Handbook. 2nd edn. O’Reilly Associates (1992)
3. Berk, E.: JLex: A lexical analyzer generator for Java(TM). 1.2 edn. (1997)

RoboCup – An Alternative Approach to teach Informatics

Hans-Jörg Haubner, and Johannes Freudenmann

University of Cooperative Education Karlsruhe,
Erzbergerstrasse 121, 76133 Karlsruhe
{Haubner, Freudenmann}@ba-karlsruhe.de

1 Motivation

To build robots is a great challenge since many centuries. The first beginning was in Europe a robot as artificial duck by Jacques de Vaucanson in 1752, a famous French constructor of machines. Since the 20th Century robots or better Handling Systems like robotic assembly lines, are a common industrial standard in manufacturing Industry. Meanwhile intelligent mobile robot assistants cooperating with humans are focused in research and development. Human robots and androids are yet only part of movies and science fiction novels.

Therefore robotics is now an important part of research and education at universities. Previously robotic equipment was complex and expensive. Since some years small robots were developed as experimental systems for education and entertainment. The new expression edutainment was created. In the following first experiences of education with such systems are reported.

2 The Robotic Invention System

The so-called RIS (Robotic Invention System) was developed by a partnership of the MIT and the LEGO Company, well known as a manufacturer for toys. It is based on Lego standard components and a brick, the so called RCX (Robot Command Explorer). This is a microprocessor consisting of an industrial system from Hitachi with three output ports for LEDs and motors and three input ports for several sensors encapsulated alike a Lego component [1].

The system contains motors, gears, and sensors that can make the robots turn, rotate, reverse direction, start and stop, respond to their environment, and execute other robotic functions controlled by the software.

The most interesting part is a programming environment running on a PC. It consists of a graphical block oriented language to easy build and download programs for several typical tasks:

- To drive a well defined route
- To follow a black line, controlled by a light sensor

- To avoid collision in a labyrinth controlled by a touch sensor

Based on this system we can teach the basics of robotics. We get a very simple and clear introduction of actors and different kind of sensors. We can start programming immediately without taking care of special complex programming environments. The students react in a very positive way, since they know LEGO and they remember playing LEGO for themselves. Another aspect is that no two robots look alike. Every one is a little different and the students can design their own personal system.

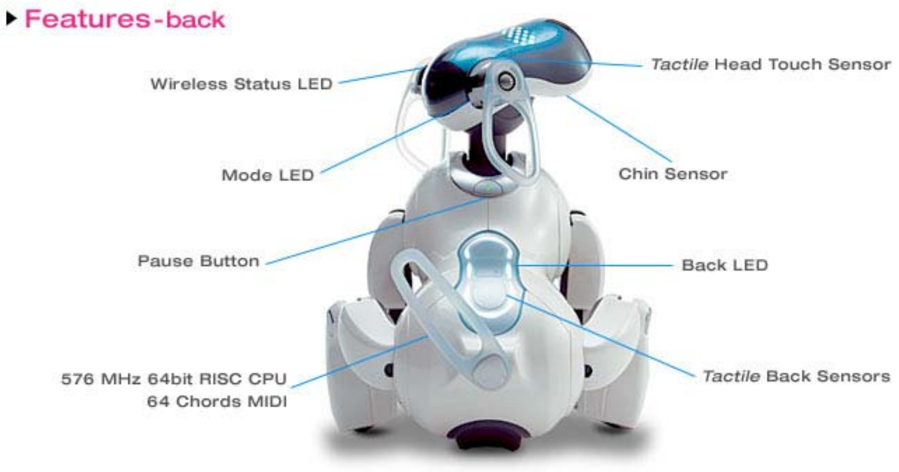
In the mean time the RCX community developed several helpful utilities and new programming environments. The most important ones are:

- A CAD System to construct three-dimensional models on a PC
- A Java based SDK with a Virtual Machine replacing the RCX firmware
- A C like programming interface
- A Basic oriented programming interface
- ROBOLAB, a powerful combination of LEGO bricks and National Instruments LabVIEW graphical development software

Based on these tools you can teach other design and programming techniques and you can deal with more sophisticated tasks. Unfortunately there are some mechanical deficiencies which prevent one to build up systems which have to run for a longer time.

3 Advanced Systems and RoboCup

In 1999 the Japanese enterprise Sony has developed with the AIBO a new advanced edutainment system, a four-legged, dog-shaped, walking robot. In addition, AIBO offers a free and downloadable software-programming tool known as Open-R SDK. These factors explain why AIBO has gained great popularity in universities around Europe where it is used to support teaching and research. Figure 1 shows the robot's sensors.



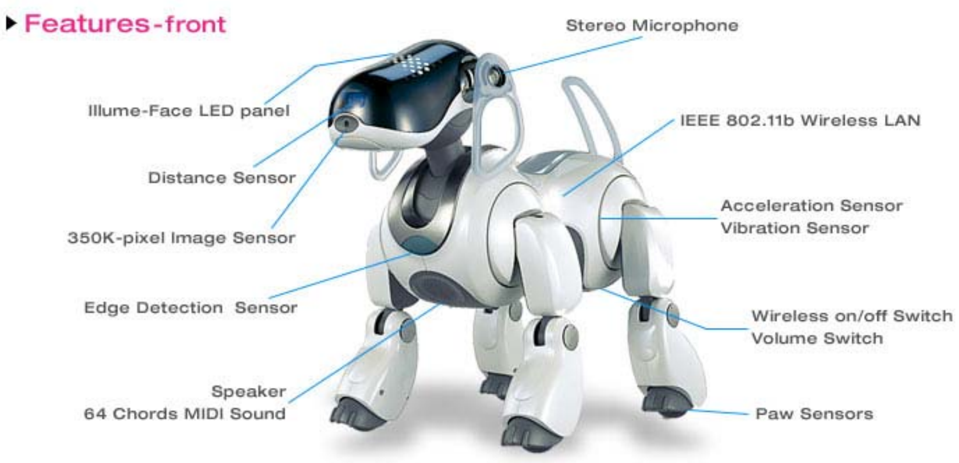


Fig. 1: Aibo ERS-7

Main features of Aibo robot

64 Bit RISC-Processor MIPS R7000 with 64 MB RAM and clock speed 576 MHz Program media is a dedicated AIBO robot "Memory Stick™" media, digital camera, WLAN IEEE 802.11b, microphone, loudspeaker, integrated web server, several sensors for measurement of distance, temperature, touch, vibration, acceleration

Movable Parts are:

- Mouth - 1 degree of freedom
- Head - 3 degrees of freedom
- Leg - 3 degrees of freedom x 4 legs
- Ear - 1 degree of freedom x 2
- Tail - 2 degrees of freedom

Built-in Sensors:

- Temperature Sensor
- Infrared Distance Sensor (head, body)
- Acceleration Sensor
- Electric Static Sensor (head, back)
- Pressure Sensor (chin, paws (4))
- Vibration Sensor

In 1997 the first championship called RoboCup was organized in Japan primarily for teams of universities. In the meantime there are several categories like simulation, small-sized, middle sized leagues and legged league. AIBO is the star at RoboCup where universities from around the globe compete in robot football competitions. A good idea gives figure 2. In between a new fascinating league is defined for humanoid robots.

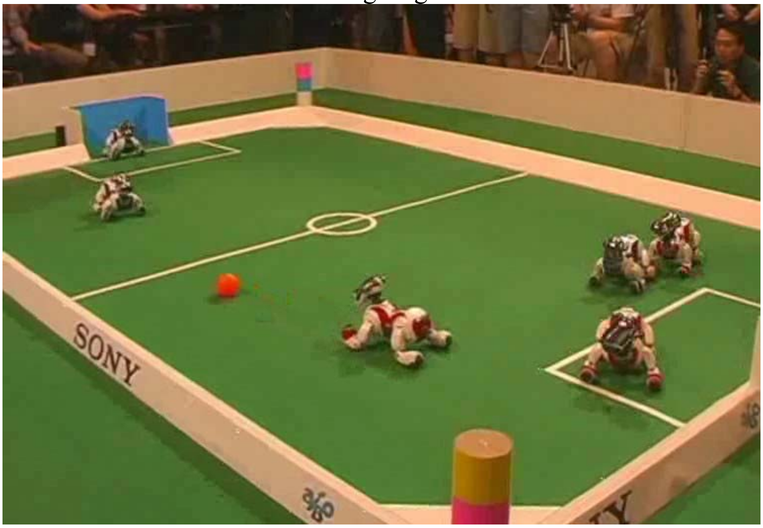


Fig. 2: Soccer match during RoboCup 2004.

We plan to set up an internal RoboCup. The idea is not to teach complex distributed algorithms by presenting the solutions, but let the students develop the solutions for their own in a guided playing manner.

For this purpose we need some more Aibos, but they tend to get cheaper and cheaper, so we think it will be affordable. We then will split a course in several groups, every group has to run through a training camp and at last they fight for the RoboCup.

Conclusion

Robots as described above are a good approach to introduce engineering concepts to students of all ages and to promote AI, robotics, and related fields. These consist of programming and technical aspects up to scientific goals like collective behavior control theory. Particularly mobile robots used as in RoboCup are a challenge and they are motivating students to develop engineering intuition. A next step could be Human vs. Humanoid. By 2050, the aim is to develop a team of fully autonomous humanoid robots to win against the human world champion team in soccer [2].

References

1. Bagnall, B.: Lego Mindstorm Programming, Prentice Hall (2002)
2. RoboCup Joint Project: www.robocup.org/

Towards a Presentation Mirror: First Steps in Using the BlueWand Technology for Gesture Analysis

Steffen Belgardt, Nico Schwan, and Dirk M. Reichardt

Berufsakademie Stuttgart, University of Cooperative Education, Information Technology,
P.O.Box 100563, D-70004 Stuttgart
steffen.belgardt@freenet.de, mail@nico-schwan.de,
reichardt@ba-stuttgart.de,
http://www.ba-stuttgart.de/dr

1 Introduction

Affective computing has become a field of research which is attracting more and more interest in computer science. Several approaches have been made concerning emotion recognition, emotion modeling, generation of emotional user interfaces and anthropomorphic communication agents. As one application Rosalind Picard proposed the “affective mirror”, which acts as an electronic teacher and companion who assists a person who is preparing for a job interview [1]. As an adaptation of this idea, we took the scenario of preparing for a *presentation*. The role of the computer is to give feedback on the performance focusing on nonverbal communication like gestures, facial expressions and vocal effects.

Current work at the Berufsakademie Stuttgart, University of Cooperative Education, focuses on analysis of vocal effects and the improvement and integration of vision based recognition of facial features and emotions [2][3]. The work presented in this paper adds a first approach to gesture recognition based on the BlueWand technology [4].

2 The Presentation Mirror

What is the basic idea behind the presentation mirror? When preparing for an important presentation most presenters would appreciate a rehearsal. Instead of doing this rehearsal with friends, colleagues or family, the presentation mirror should provide an electronic solution. What is important? The gift of oratory is not given to all of us but there are a lot of proposed general rules which we can learn in order to give good presentations. As for vocal effects these rules include adequate intonation and variation of loudness in order to stress important parts of the presentation. As for gestures Claude Cadoz [5] describes three different functions of the gestural channel: the ergotic, the epistemic and the semiotic channel. The semiotic channel describes informational messages for the environment and therefore is the one which is most interesting in our application. Natural gestures can be furthermore divided in deictic (pointing), metaphoric (suggestive), iconic (matching shape or form) and beat (emphasizing) gesture types. McNeil [6] defines a gesture unit as the period of time between rests of limbs. Identifying these units is the first task for gesture recognition. Another important finding for our task is the special role of deictic gestures in a presentation since pointing to diagrams or graphics is an often used gesture type [7]. As one general rule especially metaphoric gestures should rather be slow and spacious as opposed to quick (hectic) and small.

The gesture recognition system consists of three main phases. In phase one a segmentation takes place, which identifies gesture units. The second phase classifies these units in discrete categories. Finally, the third phase combines classified gesture units in order to interpret them and to extract semantics.

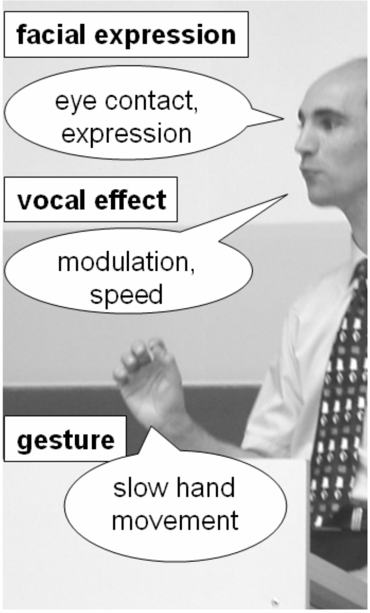


Fig. 4: The three criteria for presentation evaluation.

The feedback to the presenter consists of two levels. An abstract level feedback shows statistics done on the complete presentation showing the percentage of gesture types, static (motionless) time, the share of quick vs. slow gestures etc. The detailed level provides information on gesture type and time. This set of information can then be combined with vocal effect analysis for the same period in order to check whether both are matching.

The first steps presented in this paper focus on the use of the BlueWand technology and provide results on gesture segmentation, general slow-fast classification and first beat gesture recognition.

3 Using the BlueWand Technology

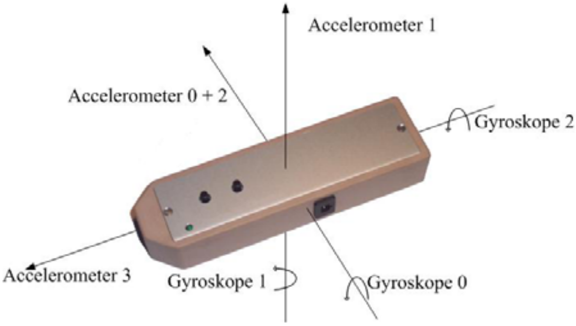


Fig. 2: The BlueWand Sensor.

The BlueWand was invented by Christian Ceelen, Markus Klein, Alexander Lange und Manuel Odendahl, Dr. Thomas Fuhrmann und Dr. Till Harbaum of the University of Karlsruhe. It was designed as a new command interface for a variety of devices which can be controlled via a Bluetooth link.

The BlueWand can be described as a generic movement sensitive input device. It may be used to analyze translational and rotational hand movements using built-in accelerometers and gyroscopes. The information from the sensors is transmitted over the built-in Bluetooth interface to client devices. The BlueWand uses the MicroBlue Bluetooth stack. Three of the seven sensors are gyroscopes recognizing the angular acceleration for each angular axis up to 300 degrees. Four accelerometers detect the linear acceleration for each dimension (the X-axis is detected twice) up to 20 m/s. Additionally, the BlueWand comes with two buttons to enable user input. A light emitted diode gives feedback about the device's different states. An additional sensor for the temperature supports the compensation of environmental impacts on the mechanic sensors.

4 First Results and Next Steps

To segment and analyze the received data several records of gestures were made. They show the acceleration for the axes during the measurements. The first graph shows a typical pointing gesture presenters do to raise attention to a slide during the presentation.

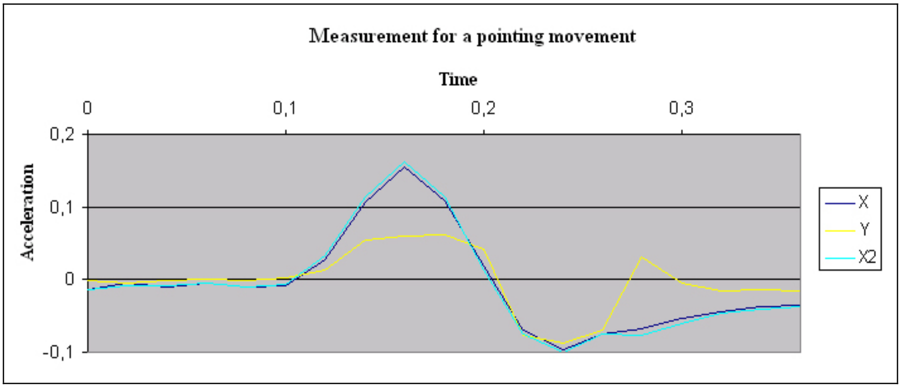


Fig. 3: Typical profile of a pointing movement.

The BlueWand was moved from the front of the presenter's to the upper right, as can be seen from the acceleration of the X and Y axis.

As a first result, we do segmentation into gesture units. For those units a classification as 'slow' and 'fast' is done.

Slow and fast movements are illustrated in the next figure. If the BlueWand is moved fast, the amplitude of the acceleration is higher than for slow movements. For the next step the red graph shows a segmentation of the acceleration in three different classes.

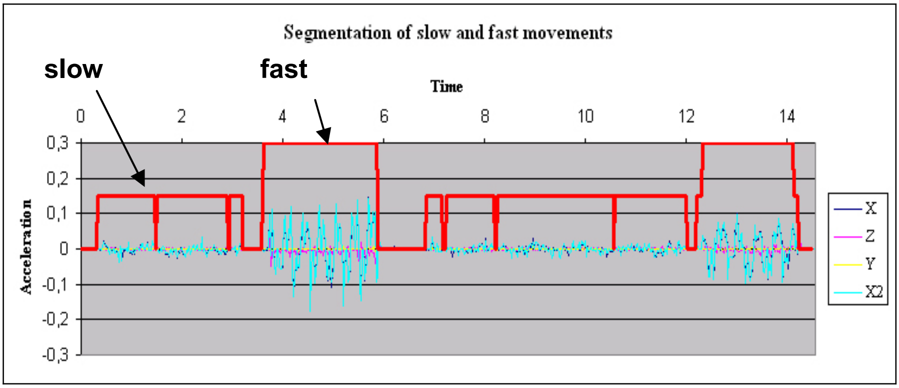


Fig. 4: Segmented measurement of slow and quick gesture units.

Current work focuses on the classification of pointing gestures, beat gestures and iconic gestures. For iconic gestures we identify gesture primitives and allow the abstract definition of complex gestures (shapes like boxes, etc.) by a rule set.

5 Conclusion and Acknowledgements

Future work, supported by student research projects, will concentrate on the recognition and classification of further gesture types as well as on the integration with vocal effect analysis and facial expression classification. Furthermore we want to thank Alcatel Research and Innovation for providing the BlueWand technology.

References

1. Rosalind W. Picard: Affective Computing. MIT Press 1997
2. Dirk M. Reichardt: "Applying Computer Science Methods to User Emotion Recognition", in: Weghorn (ed.), Proceedings of the 1st Annual Meeting for Information Technology & Computer Science at the BA-University of Cooperative Education, Stuttgart, 2004
3. Stefanie Maoro, Dirk M. Reichardt: "Approaches to Vision Based Emotion Recogition Using Neural Networks", in: Weghorn (ed.), Proceedings of the 2nd Annual Meeting for Information Technology & Computer Science at the BA-University of Cooperative Education, Stuttgart, Germany, 2005
4. Thomas Fuhrmann, Markus Klein, Manuel Odendahl, “Bluewand - A versatile remote control and pointing device”, Proceedings of the 13th ITG/GI-Fachtagung Kommunikation in Verteilten Systemen (KiVS'2003), Kurzbeiträge, Praxisberichte und Workshop, pages 81-88, 25.-28. February 2003, Leipzig, Germany
5. Claude Cadoz. “Le geste canal de communication homme/machine ". In : Technique et Science Informatique, Vol. 13, No 1, pp. 31-61, 1994.
6. D. McNeill, “Hand and Mind: What Gestures reveal about thought”, University of Chicago Press, 1995
7. J. Eisenstein and R. Davis. “Natural Gesture in Descriptive Monologues”. 2003 ACM Symposium on User Interface Software and Technology (UIST'03).

Integration Engineering: Construction of Software Process Models towards the Implementation of Inter-Enterprise Value Chains

Antonius van Hoof

Berufsakademie Stuttgart, Außenstelle Horb, Florianstr. 15,
72160 Horb/N., Germany
a.vanhoof@ba-horb.de

Abstract. Present day cooperation of enterprises to establish new markets call for the construction of inter-enterprise value chains. Apart from strategic and organizational problems involved, this need also translates into the problem of the design and construction of a software infrastructure to technically implement these value chains. The technical aspect of the latter problem seems to point into the direction of Service Oriented Architectures, with their use of internet technologies. The present paper tackles the organization problem of actually designing and constructing such technical solutions. It deals with the construction of software process models for the implementation of inter-enterprise value chains. It is shown, that such process models in having to account for internet-speed development, i.e. negotiating quality and functionality for as quick as possible delivery, inherit central features of so-called agile processes. A concrete adaptable software process model is presented.

1 Introduction

Industry today has identified many opportunities for joint cooperation in many markets, but with that, it also faces the problem of having to integrate value chains across multiple companies[2]. Apart from the enormous strategic and organizational problems involved, there is also a large technical problem to be solved: the practical integration of the IT-infrastructure, present at each company, into the technical super-infrastructure, meant to support the processes, applications, and data involved in the envisaged value chain. Whereas with the upspring of Service Oriented Architectures the technologies involved to handle inter-enterprise application integration seem to have emerged, the question remains, how to tackle projects to practically implement such architectures in the face of existing heterogeneous applications and platforms. This is a problem in the field of Software and Systems Engineering: the construction of software process models toward

[2] This paper presents some preliminary results from a research project called "Intergration Engineering", funded by the German Bundesministerium für Bildung und Forschung as part of the research programme "Software Engineering 2006".

the implementation of inter-enterprise value chains. In the context of the present work, we will refer to this as the integration engineering problem.

2 Main Characteristics of Integration Engineering Projects

Projects that tackle the integration engineering problem will be called integration engineering projects (IE projects for short). What characterizes such projects is:

- *Internet speed development;* Success in business integration lies in the speed of the implementation of the new value chains.
- *Disperse development;* Overall architecture and development of parts of the solution (i.e. service modules) have to be designed and implemented by disperse teams, used to do their work according to the rules and structures of their own respective companies. Software process organization can be totally different between separate organizations with respect to functionality, tool support and maturity.
- *Interoperability handling;* In most cases, the lump sum functionality needed for the implementation of the inter-enterprise value chain is already present in disperse applications at the premises of the value chain partners. The problems facing the developers is now, to make each application behave as a part of a greater whole, i.e. to abstain from central control, and to cater for emergent system properties, i.e. to ensure system-of-systems stability and functionality in the face of the possible situation, that its behavior is more than the sum of its independent applications.

In the following we will focus on the consequences of internet speed development. For the other aspects mentioned the reader is referred to, amongst others, [8, 11].

2.1 Internet Speed Development

Literature provides ample material to substantiate a "State of the Art" of Software Engineering. Much of this is embodied in the Rational Unified Process RUP [7, 9]. In how far this state of the art can also be counted as a "State of the Practice" can be doubted. See, e.g., Cusumano, M., MacCormack, A. et al. [4] for a world-wide investigation of current practices. In the same vein, In an article [1], Baskerville, R., Balasubramaniam, R. et al. have focused on the question whether the software development of internet applications is different from "traditional" (in the sense of State of the Art) software development. They come to the conclusion, that internet applications are heavily driven by the time perspective, in such a way that quality and functionality of the software are negotiable against speedy introduction of the software. They conclude that this kind of software development has more of the properties of agile software development processes in the sense of Beck's XP or Cockburne's Crystal (see [2, 3, 5, 18]). As our type of project, IE-projects, can be subsumed under their definition of internet-speed project, IE-projects have to consider following aspects with full priority:

- They must be able to react to fast changing an instable user requirements;
- They must assure an intensive and frequent contact with the users;
- They need to provide frequent software releases and prototypes to the users;
- They must provide these releases in time.

As a consequence, software process models for IE will have to orient themselves towards agile development. How these kind of software process models can be accommodated into RUP is shown e.g. in [9, 10]. A dynamic link library (dll) is a memory saving variant of a link library. Whereas a static link library transfers the source code of any call into the application, a dynamic link library uses dedicated links. This means that a dynamic link library loads the code into the RAM just once, thus saving a large amount of memory and data space.

3 Constructing Software Process Models for IE

In constructing software process models, one has to consider a lot of things, that all interrelate as shown in figure 1.

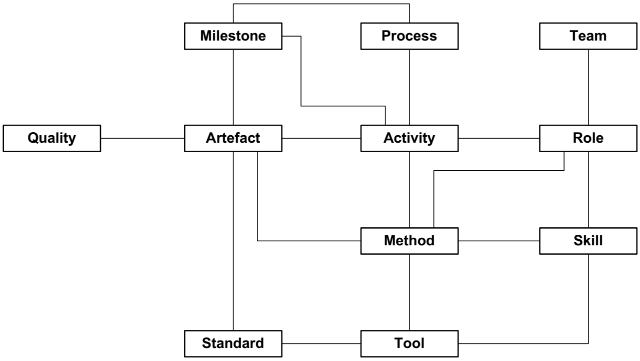


Figure 1: Software process meta-model (after [2], page 257)

In the following we will only focus on the activity part of the software processes. Figure 2 presents the central part of a generic process model in the form of a UML activity diagram. It is embedded in an iteration (not shown here) that has to be followed through in short intervals of between 4 to 10 weeks depending on the length of the project (normally between 3 to 12 months). The result of each iteration is a software release.

3.1 Component Engineering, Ontological Engineering

The activities shown in Figure 2 are on the one hand concerned with component engineering processes as described in Weisbecker [17]. On the other hand, they are concerned with enriching software analysis and design with ontological engineering activities to ensure both, that all companies involved in the inter-enterprise value chain focus on the intended services functionalities, and that the implemented services can be reused more easily in the course of implementation of future value chains. Standard modeling technologies (see e.g. [14, 15, 16, 19]) for ontological engineering are being used.

Since IE-project focus upon technical implementation in the context of service oriented architectures, modeling technologies as developed by the OASIS consortium (see [12]) are used.

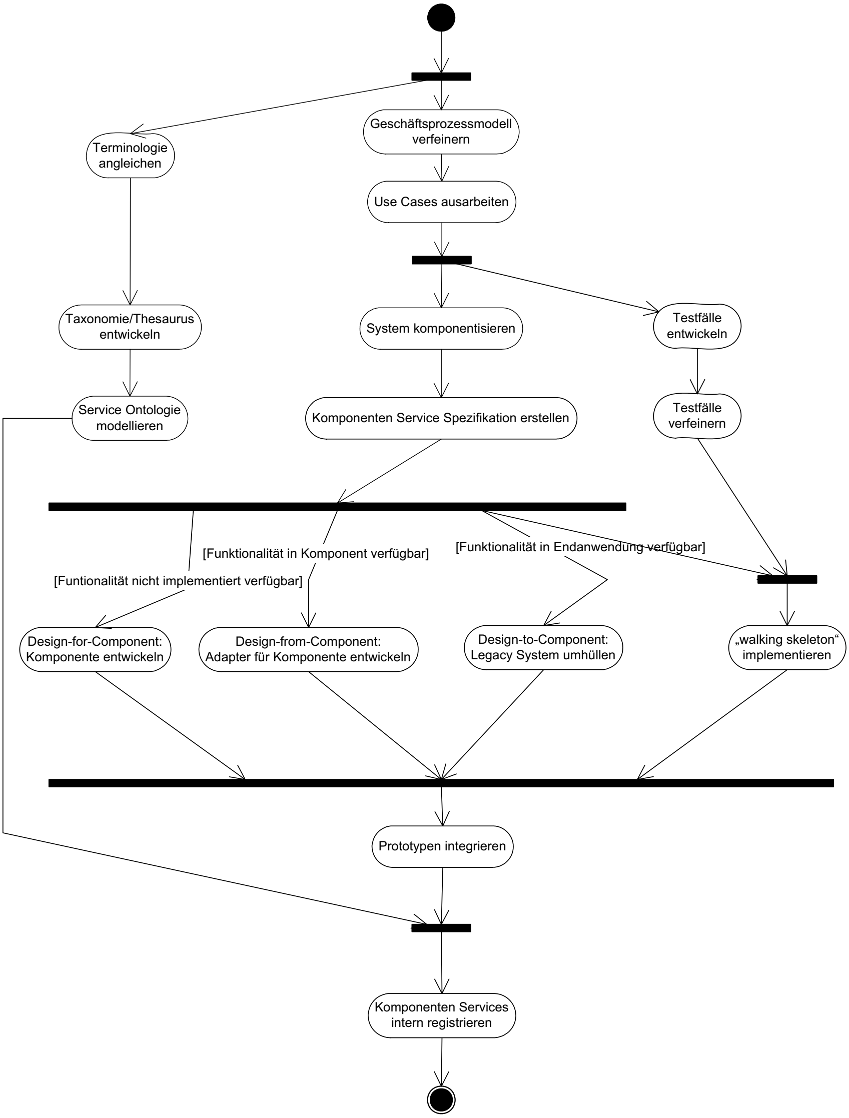

Geschäftsprozessmodell verfeinern
Terminologie angleichen
Use Cases ausarbeiten
Taxonomie/Thesaurus entwickeln
System komponentisieren
Testfälle entwickeln
Service Ontologie modellieren
Komponenten Service Spezifikation erstellen
Testfälle verfeinern
[Funktionalität in Komponent verfügbar]
[Funktionalität in Endanwendung verfügbar]
[Funtionalität nicht implementiert verfügbar]
Design-for-Component: Komponente entwickeln
Design-from-Component: Adapter für Komponente entwickeln
Design-to-Component: Legacy System umhüllen
„walking skeleton“ implementieren
Prototypen integrieren
Komponenten Services intern registrieren

Figure 2: Central part of a project iteration in a generic software process model for IE

References

1. Baskerville, R., Balasubramaniam, R. et al., “Is Internet-Speed Software Development Different?” in. IEEE Software (20), Nov./Dec. 2003, pp. 70-77
2. Cockburn, A., Just-In-Time Methodology Construction, http://alistair.cockburn.us/crystal/articles/jmc/justintimemethodologyconstruction.html, referenced: 2004-08-20
3. Cockburn, A., Crystal Clear. A Human-Powered Methodology For Small Teams, including The Seven Properties of Effective Software Projects, http://alistair.cockburn.us/crystal/books/cc/crystalclear.zip, referenced: 2004-08-20
4. Cusumano, M., MacCormack, A. et al., “Software Development Worldwide: The State of the Practise” in. IEEE Software (20), Nov./Dec. 2003, pp. 28-34
5. Fowler, M., The New Methodology, http://www.martinfowler.com/articles/newMethodology.html, referenced: 2004-08-20
6. Gawlick, D., “Infrastructure for Web-based Application Integration”, in: Proceedings of the 17th International Conference on Data Engineering (ICDE’01), IEEE Computer Society, 2001, 4 pages
7. Kruchten, Ph., The rational unified process: an introduction, Addison-Wesley, 1998
8. Morris, E., Levine, L. et al., System of Systems Interoperability (SOSI): Final Report, SEI Technical Report, CMU/SEI-2004-TR-004, Software Engineering Institute, Pittsburgh, 2004
9. Pollice, G., Using the Rational Unified Process for Small Projects. Expanding upon eX-treme Programming, Rational Software White Paper, TP183, Rational Software Corpora-tion, 2001, 15 pages
10. Smith, J., A Comparison of RUP and XP, Rational Software White Paper, TP167, Ra-tional Software Corporation, 2001, 25 pages
11. Tolk, A., Muguira, J.A., "The Levels of Conceptual Interoperability Model", in Proceed-ings of the 2003 Fall Simulation Interoperability Workshop, Orlando, Sept. 14-19th, 2003
12. UN/CEFACT Modelling Methodology (CEFACT/TMWG/N090R9.1), 2001
13. W3C, Resource Description Framework (RDF): Concepts and Abstract Syntax, Recommendation 10 February 2004, http://www.w3c.org/TR/2004/REC-rdf-concepts-20040210/, referenced: 2004-06-29
14. W3C, RDF Primer, Recommendation 10 February 2004, http://www.w3c.org/TR/2004/REC-rdf-primer-20040210/, referenced: 2004-06-29
15. W3C, RDF Vocabulary Description Language 1.0: RDF Schema, Recommendation 10 February 2004, http://www.w3c.org/TR/2004/REC-rdf-schema-20040210/, referenced: 2004-06-29
16. W3C, OWL Web Ontology Language Guide, Recommendation 10 February 2004, http://www.w3c.org/TR/owl-guide/, referenced: 2004-06-29
17. Weisbecker, A. (Hrsg.), KoSPuD. Komponentenbasierte Software für Produkte und Dienstleistungen, Abschlussbericht, Stuttgart, 2000, 216 pages
18. Wells, D., Extreme Programming, http://www.extremeprogramming.org/, referenced: 2004-08-12
19. Yao, H., Etzkorn, L., “Towards A Semantic-based Approach for Software Reusable Component Classification and Retrieval”, ACMSE’04, April 2-3, 2004, Huntsville, Alabama, ACM press, S. 110-115

Author Index

List of Participants

Achim Dörnen, Universität Stuttgart, a.doernen@physik.uni-stuttgart.de
Andreas Walden, Philips Medizin Systeme Böblingen
Antonius van Hoof, Berufsakademie Stuttgart-Horb, a.vanhoof@ba-horb.de
Axel Richter, Berufsakademie Stuttgart
Axel Sikora, Berufsakademie Lörrach, sikora@ba-loerrach.de
Beate Bossler, Berufsakademie Stuttgart, bossler@ba-stuttgart.de
Christian Grünler, Berufsakademie Stuttgart / Staffordshire University, c.d.grunler@staffs.ac.uk
Dirk Reichardt, Berufsakademie Stuttgart, reichardt@ba-stuttgart.de
Dirk Saller, Berufsakademie Mannheim, saller@ba-mannheim.de
Eckhart Hanser, Berufsakademie Lörrach, hanser@ba-loerrach.de
Hans Weghorn, Berufsakademie Stuttgart, weghorn@ba-stuttgart.de
Hans-Jörg Haubner, Berufsakademie Karlsruhe, haubner@ba-karlsruhe.de
Hans-Jürgen Herrler, Berufsakademie Stuttgart
Holger Haas, FreiNet GmbH, Freiburg, haas@freinet.de
Jan Altenberg, Homag AG, Schopfloch, jan.altenberg@gmx.de
Jens Niemann, T-System Internat.GmbH, Leinfelden-Echterdingen, jens.niemann@t-systems.com
Joachim Schmid, Berufsakademie Karlsruhe, Joachim.schmid@berthold.com
Jochen Elzmann, Berufsakademie Stuttgart, elz@ba-stuttgart.de
Johannes Freudenmann, Berufsakademie Karlsruhe, freudenmann@ba-karlsruhe.de
Karl Friedrich Gebhardt, Berufsakademie Stuttgart, kfg@ba-stuttgart.de
Konrad Reif, BA Ravensburg-Friedrichshafen, reif@ba-ravensburg.de
Martin Plümicke, Berufsakademie Stuttgart-Horb, pl@ba-horb.de
Michael Müller, PROTEC MIS, Oberstenfeld, michaelmueller1976@gmx.de
Midhat Mesic, Zoller GmbH & Co. KG, Freiberg/N., mesic@zoller-d.com
Nico Schwan, Berufsakademie Stuttgart, nschwan@web.de
Olaf Herden, Berufsakademie Stuttgart-Horb, o.herden@ba-horb.de
One anonymous participant, Böblingen
Paul Freyer, Zoller GmbH & Co. KG, Freiberg/N., freyer@zoller-d.com
Rudolf Messer, Berufsakademie Stuttgart, messer@ba-stuttgart.de
Sabrina Riedmüller, BA Lörrach / Bucher Hydraulics GmbH Klettgau, s_riedmüller@web.de
Sebastian Mezger, Berufsakademie Stuttgart / Agilent Technologies Böblingen
Sebastian Ries, Berufsakademie Stuttgart / Agilent Technologies Böblingen
Sebastian Wagner, Agilent Technologies, Böblingen, Sebastian_wagner@agilent.com
Stefanie Maoro, Berufsakademie Stuttgart / HP Böblingen, steffi.maoro@freenet.de
Steffen Belgradt, Berufsakademie Stuttgart / Alcatel S.-Weilimdorf, steffen.belgardt@freenet.de
Thilo Bauer, BWMC, St. Augustin, tbauer@bwmc.net
Uwe Brecht, Berufsakademie Lörrach / MIMOT GmbH Lörrach, brecht.uwe@gmx.de
Werner Sinz, Berufsakadmie Stuttgart