

## Procedural Creation of Medical Reports with Hierarchical Information Processing in Radiation Oncology

Martin Vogel<sup>2</sup>, Harald Fahrner<sup>1</sup>, Mark Gainey<sup>1</sup>, Stefan Kirrmann<sup>1</sup>, Felix Heinemann<sup>1</sup>

*1 Klinik für Strahlenheilkunde, Department für Radiologische Diagnostik und Therapie, Universitätsklinikum Freiburg*

*2 Steinbeis Forschungszentrum für Medizintechnik und Biotechnologie, Weinheim*

Correspondence: Dr. med. Felix Heinemann, Klinik für Strahlenheilkunde, Department für Radiologische Diagnostik und Therapie, Universitätsklinikum Freiburg, Robert Koch Str. 3, 79106 Freiburg,

Tel. +49 761 270-94550, Fax +49 761 270-95520

E-Mail: [felix.heinemann@uniklinik-freiburg.de](mailto:felix.heinemann@uniklinik-freiburg.de)

**Abstract Background:** For many years, the oncological doctor's letter has been the pivotal means of information transfer to general practitioners, medical specialists or medical consultants. Yet, both creator and recipient require a high level of abstraction, retentiveness and analysis due to the large number of diagnoses and therapies. In contrast to the commonly used structure of doctor's letters, where all diagnoses and therapies are listed in sequential order with all diagnoses first, it is by no means trivial to establish the important chronological and hierarchical context in the description of oncological cases. Additional aspects of importance are the integration of these letters into existing clinical and departmental information systems (for example via HL7 interface), various export formats (for example PDF, HTML), fax and encrypted email. Moreover these letters need a modern layout that, among others, meets the requirements of corporate design. **Methods:** The requirements for a doctor's letter system are manifold and can only be represented rudimentarily via a normal word processing system. Due to this deficiency we developed a system that covers all special features and requirements for clinical use. The system is based on a scalable and extensible client-server architecture. We use the programming languages Harbour, C++, PHP and JavaScript, Microsoft SQL database for data storage and the HL7 standard as the interface to other information systems such as hospital information system (HIS). Export formats are PDF, HTML/XML. Layouts are generated with TeX, LaTeX and MikTeX. **Results:** The aforementioned requirements were resolved with the doctor's letter and finding system IntDok. The hierarchical presentation of diagnoses, histologies and therapies provides the recipient with a first outline of the course of the disease. A strict procedure controls the whole process of document compilation and assists the user with many highly regarded tools such as text blocks, import and export (PDF and HTML/XML including barcodes) functions or HL7 interface to other information systems. The software also provides a sophisticated mail merging. All content from previous letters can easily be inserted into the current document. A TeX-server automatically provides document layout including supreme hyphenation so that uniform and perfect appearance (corporate design) is guaranteed. The documents are saved in a MS-SQL database (almost 230,000 documents since 1991), independent of any proprietary formats such as MS-Word. **Conclusion:** Creation of documents is fast, simple and well-structured. Sophisticated tools guarantee the optimal use of human resources and time. The system is an important module in our overall digital work environment.

*Keywords:* Doctor's letter, clinic management, workflow, quality management


### Background and Subject

For many years, the oncologic doctor's letter is the pivotal means of information transfer to general practitioners, medical specialists or medical consultants. As the personal computer found its way into the standard business environment since the mid-1980s, office software – WordStar, WordPerfect, followed by Microsoft-Word and OpenOffice – were brought to the market. These software packages slowly replaced the traditional way of writing doctor's letters and reports of medical findings with a typewriter.

Both creator and recipient of an oncologic doctor's letter require a high level of abstraction, retentiveness and analy-

sis due to the large number of diagnoses and therapies. In contrast to the commonly used structure of doctor's letter, where all diagnoses and therapies are listed in sequential order with diagnoses first, it is by no means trivial to establish the important chronological and hierarchical context in the description of oncological or traumatological cases. These circumstances were pivotal in the development process of the software, which will be shown here. The project started in 1991 and is still being developed.

The integration of the documents into existing hospital information system (HIS, for various abbreviations see Table 2 in the appendix) or departmental information sys-

This article is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA. [Creative Commons Attribution 4.0 International license.](#) 

tems via HL7-interface [1-3], various export formats (PDF, HTML/XML or TIFF), fax and encrypted email transmission, a modern layout based on TeX [4-6], to the end of meeting the criteria of corporate design [7] were additional aspects of importance. All documents should be available online [8].

Furthermore, layout and content should be easily adjusted to new requirements. Documents should be stored in a database [9, 10] to maintain platform independence and to fulfil the requirements set by radiation safety legislation [11]. With specific barcodes on each document it becomes possible to automatically distribute the documents to other systems [12]

Finally, another important aim is defined by the effective use of human resources to create the optimal doctor's letter. Powerful software tools are needed to that end, which are presented in this work as well.

A central requirement for the system presented here is the integration into our digital concept, which has existed for many years [13-15].

## Material and Methods

The requirements for a doctor's letter system are manifold and can only be represented rudimentarily via a typical word processing system.

Due to this failing, we developed a system (IntDok) that covers all special features and requirements for clinical use. In addition, our current departmental system MOSAIQ (version 2.6, Elekta AB Sweden) does not meet our requirements for writing doctor's letters.

Because of the complexity of the task, a thorough analysis of the requirements was carried out prior to commencement and implementation, where the authors could draw from a wealth of experience. In this context we have seen that there are only a few commercial systems that attended to this matter. Thus, standard programs such as Microsoft Word were and are the tools used for writing doctor's letters.

Table 1 depicts the requirements that resulted from the analysis mentioned before, including the corresponding year of each step of implementation.

Table 1  
*Requirements*

Year	Requirement
	Development environment:
1991	<ul style="list-style-type: none"> <li>• dBase (vendor: Ashton Tate/Borland, USA)</li> <li>• Clipper (vendor: Nuntucket, USA)               <ul style="list-style-type: none"> <li>○ compiler version of dBase compatible programming and database system</li> </ul> </li> <li>• C++ language (vendor: Borland, USA)</li> </ul>
1991	Development of the Medatec Structure Definition Language (MSDL), including the corresponding interpreter/compiler
1991	Development of an interactive Tool, for generating input masks. Interactive Mask Generator (IMAGE)
1991	Strictly sequential order of all procedural steps
1991	Mapping of the standard clinical case-orientated structure of clinical departments
1991	Access rights management
1991	Interface to the hospital information system (HIS)
1991	Saving documents in a database (Clipper/dBase)
1991	Mail merge
1991	Reusable text blocks / elements
1992	Import functions
1992	Hierarchical layout of diagnoses, histologies and therapies in one block
1995	BDT interface to electronic insurance card
1995	Migration of Clipper database to advantage database server ADS (vendor: Extended Systems/SAP), still dBase compatible, but with a client server architecture
1995	Start of migration from MS-DOS to Windows 3.1x
1997	End of migration from MS-DOS to Windows 3.1x
1998	Development of the Type Programming Language (TPL), including the corresponding interpreter
1999	Development of TeX-file enhancing Language (TXF) , including the corresponding interpreter
2001	Implementation of a TeX server
2002	Support added for relevant file formats like PDF, HTML, XML
2003	Implementation of a web server for platform independent visualization of the documents
2003	Streamlining of the software administration

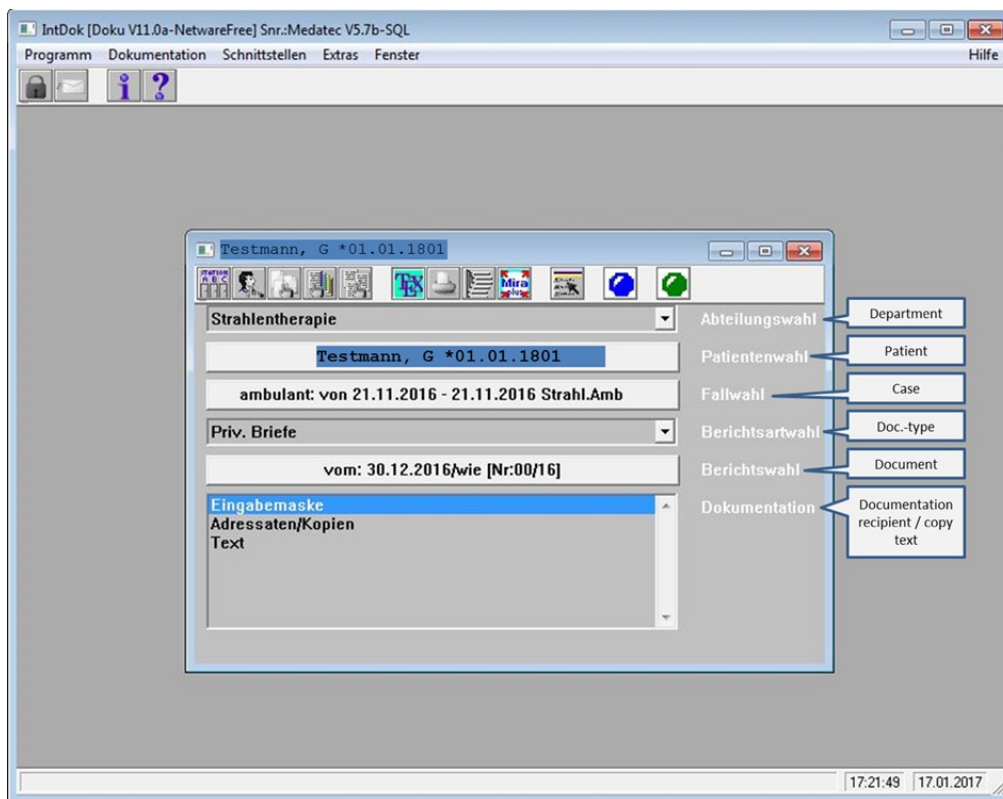
2003	Improvement of configuration options
2004	Improved functionality for import and export
2005	Integration of graphics and images
2007	Redesign with improved modularity
2008	Migration to a MS-SQL database (vendor: Microsoft, USA)
2009	Migration from Windows 3.1x to the 32Bit platform Harbour (open source) for Windows (32Bit xBase compiler)
2010	Integration of speech recognition
2010	Integration of barcodes
2014	Standardized export in PDF format via proprietary interface
2016	Standardized export in PDF format via multifunctional, standardized interface

The following tools have been used to realize the project:

- Programming languages Harbour (xBase, open source) for Windows with C++ (Borland)
  - is the main programming environment for all Windows programs/modules
- TeX, LaTeX, MikTeX Scripting language
  - is used for the layout design of documents
- Database MS-SQL by Microsoft
  - Storage system for all accruing data
- Three scripting languages developed in-house
  - explained below
- PHP (Hypertext Preprocessor), JavaScript, JQuery, (all open source)
  - is used for all web based operations

A detailed description of all functionality is beyond the scope of the present work, so that the following paragraphs are limited to the most important functions. IntDok is a module of the parent system MEDATEC, which is explained later in the section technical background. The system has been, and is being, continuously developed in close cooperation with the users. The cooperation with the user was, and still is, a decisive factor for the constant improvement and extension of the system. Above all, the users recognize the strengths and weaknesses of a program best during their daily work.

Functionality in the period until 1995: The creation of documents was workflow optimized and enabled efficient working. Operation was simple and the guidance through the program is strictly procedural so that even new users readily familiarize themselves with the system.



**Figure 1.** Start screen of IntDok (document producer and handler)

From the outset, we have ensured that documents are created with clinical cases in mind – as it is typically and usually done in clinics (Figure 1) – and in accordance with a comprehensive access and control rights concept.

As early as 1991, a first software version running with the operation system MS-DOS (Microsoft, Redmond) was established in our department which already met the most important requirements. It is noteworthy that, from the beginning, the project included a pseudo-graphical interface that we named ‘ergonomic shell’ (‘ErgoShell’).

In addition, we attached great importance to store all documents within a database, independent of any proprietary file formats as, e.g., MS-Word. We started in 1991 with the Clipper compatible programming environment.

Furthermore, the software offers a sophisticated mail merge system with one main addressee and up to 10 recipients in carbon copy. The software then automatically creates copies for primary care and archive for example. The address field of all copies contains the respective address with automatic adjustment of the salutation (e.g., Mr. or Ms.); manual copying of letters and tedious addressing can be completely omitted.

Import of demographic and clinical case-related data from the hospital information system (HIS) has been implemented in this time interval. As content such as diagno-

ses, therapies or other assessments can be imported from a patient’s previous documents, it eliminates repetitive input of existing data. User-defined text blocks essentially simplify the creation of documents.

An essential core functionality was to combine diagnoses, histologies and therapies in a hierarchical way (Figure 2). This combination ensures that each therapy (and each histology, if needed) is directly linked to the corresponding diagnose. A tree structure is created with increased comprehensibility that immediately provides the reader with a clear and meaningful overview about recent diagnoses and therapies (Figure 3).

Figure 2 shows a typical input form for an oncological doctor’s letter. The secretary fills in the corresponding fields and the letter is then automatically generated based on this data via the corresponding TeX template. Behind most of the fields sub-forms are hidden, which can be reached via the small button to the right of a field. Depending on the requirements, various functionalities are available there. For example, the button right to the field ‘recipient’ leads to a multiple selection list-box to select the different recipients.

Integration of an interface to BDT (Behandlungsdaten-transfer = transfer of treatment data) enabled the use of patient’s electronic insurance cards in 1995; BDT is the

**Figure 2.** Data input form for writing a doctor’s letter.

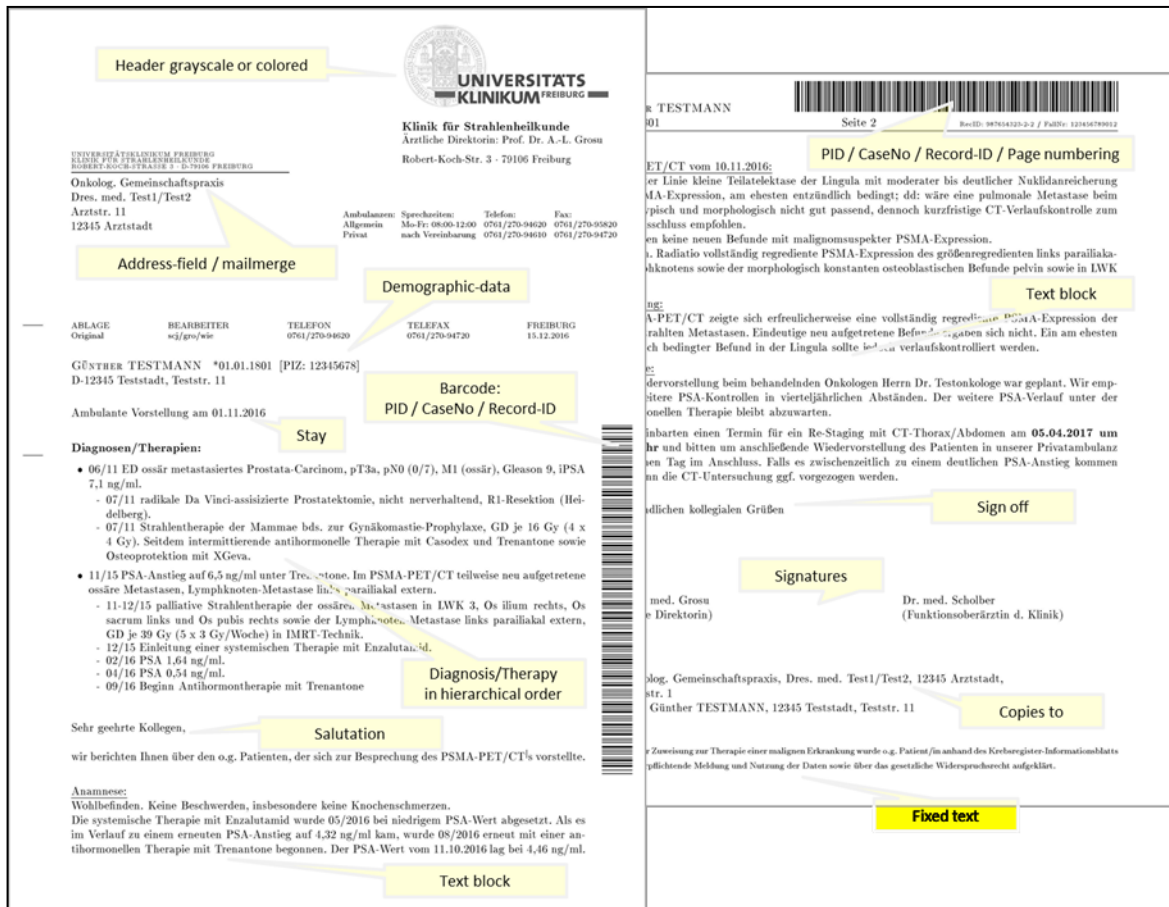


Figure 3. Example of an oncological doctor letter using TeX Layout.

official standard of the federal Health Insurance Association for the exchange of treatment data in Germany. Import of demographic data stored on those cards was possible, rendering the system even more interesting for external practitioners.

**Functionality in the period from 1995:**

**1995 to 1997:**The whole software package was migrated from MS-DOS to Windows 3.1x, facilitating functionality like copy-paste from other programs, or the inclusion of images and graphics; the latter functionality, however, was only in full use from 2005 onward.

**2001:** Since 2001, documents are rendered with a custom-built TeX-Server [4-6] and special templates, so that a thoroughly homogenous and professional corporate design is ensured (Figure 3).

Document authors choose data either from predefined (yet customizable) lists or fill in text fields (Figure 2). The remaining steps including hyphenation and spell checking are carried out by the system. Spell checking is extensible so that medical terms or labels of (generic) drugs are easily added.

In addition, depending on access rights of the user, the

finalized document can be locked to prevent further editing.

**2002:** Support added for relevant file formats like PDF, HTML and XML

**2003-2005:** In 2003, we introduced a web server (Web-Doc), developed in-house [8] to make all documents accessible online (Figure 4); with regard to data protection, this service is only available within our secured network (Intranet). In addition, this web server offers other medical documents and findings related to a patient (Figure 5). For this functionality, we have programmed a proprietary interface to access the documents via HTML format (added 2002). We streamlined the software administration, improved configuration options as well as import and export abilities. In addition we enabled the integration of graphics and images.

**2007:** At first, the software architecture was fundamentally re-designed, and modularity was decisively enhanced, so that new requirements could be implemented even easier.

Thus, the system can be used in virtually all medical disciplines, and it is multi-client enabled, serving, for example multiple departments within one clinic. The software provides a high level of configurability and, in addition to standard doctor’s letters, clinical reports (for example about surgical procedures), medical documentation or miscella-

neous documents or correspondence can be created.

In 2008 the database was migrated to a MS-SQL [9, 10] database, which is still in use today. It should be noted that all of about 230,000 documents created since 1991 have been successfully migrated despite numerous system updates, underlining the future-proof storage concept.

**2009-2010:** In this period, we completed the migration from 16-bit to 32-bit operating system.

**2010:** A speech recognition program (Dragon) was added in 2010. Due to the intervention of the staff council we had to deactivate this feature after 2 months. Since Windows 7, speech recognition has been integrated into the operating system and allows direct dictation in any Windows text box; therefore external recognition software is no

longer required, but can be easily integrated and used in the same way.

All documents are provided with a barcode on each page containing the patient identifier, document quality, number of pages, current page number and clinical case number (Figure 3).

**2014-2016:** Barcodes gave us the opportunity to automatically distribute the documents to other systems via our self-developed import / export tool [12]. As an example: Finalized doctor's letters from our department are sent simultaneously as a PDF, fully automatically and categorized (for example doctor's letter or finding to the hospital information system (HIS), as well as to our departmental system (MOSAIQ™)

Department für Radiologische Diagnostik und Therapie  
 KLINIK FÜR STRAHLENHEILKUNDE  
 Ärztliche Direktorin  
 Universitätsprofessorin  
 Dr. A.-L. Grosu

UNIVERSITÄTSKLINIKUM FREIBURG  
 Klinik für Strahlenheilkunde ROBERT-KOCH-STR.3 D-79106 FREIBURG

Frau  
 Dr. med. Marga Testfrau  
 Testerstr. 34  
 D-79106 Freiburg

ABLAGE	BEARBEITER	0761/270-3862 TELEFON	0761/270-3982 TELEFAX	FREIBURG
Original	hen/guz			17.09.2002

David Zaubermann \*01.01.1423  
 D-79106 Freiburg, Hugstetterstr. 23

Vorstellung am 12.09.02

Diagnose/Therapie:  
 / 08/02 Zweitcarcinom der Zunge links T3, PEC, G3, [ICD10: C02.1L]  
 sowie Plattenepithelcarcinom der Rachenhinterwand T1, gering  
 verhornendes Plattenepithelcarcinom [ICD10: C10.3] nach  
 Hypopharynxcarcinom pT2 pN0 1987.  
 - 12/87 Laryngektomie und Neck dissection.  
 - 01-03/88 postoperative Strahlentherapie, GD 60 Gy.

Sehr geehrte Frau Kollegin,

besten Dank für die freundliche Überweisung des Patienten Herrn  
 David Zaubermann zur Beratung vor einer in erster Linie palliativen  
 Strahlentherapie des Zungenrandes/Zungengrundes sowie perkutan des

**Figure 4.** Screenshot of a doctor's letter using the web service interface.

### Technical background

Finally, we would like to outline the technical background of the system. Due to the complexity of this topic, we cannot present this in detail in this paper.

IntDok is a module of the parent system MEDATEC (Medical Data and Text processing with Computer). IntDok

was the very first module we developed inside this environment. Since the developments took place in parallel, a logical separation between the two systems, at least at the beginning, was not readily apparent.

As mentioned above, custom scripting languages have been developed for this project.

**Bitte treffen Sie eine Wahl:** Please make a selection

Host: Zaubermann, David \*01.01.1423 [ID: 12345678]

- [IntMed-Dokumentenview](#) — doctors letters
- [HydMedia-Archiv](#) — patient demographics
- [Patienten-Stammdaten](#) (via PDV) — patient treatment data
- [Patienten-Bestrahlungsdaten](#) — lab values
- Laborwerte (ZLab via MiraPlus) — today
  - current
    - o [aktuellste](#) / Tag(e): [heute](#) [bis -1](#) [bis -3](#) [bis -5](#) [bis -9](#)
      - up to 9 days ago
      - up to 3 weeks ago
      - up to 3 months ago
  - list
    - o Liste (Tage): [heute](#) [bis -1](#) [bis -3](#) [bis -5](#) [bis -9](#) [bis -2Wo](#) [bis -3Wo](#) [bis -1Mo](#) [bis -2Mo](#) [bis -3Mo](#)
- Pathologie-Befunde (Patho via Miraplus) — pathology findings
  - o [aktuellster](#) / Jahr(e): [bis -0.5](#) [bis -1](#) [bis -2](#) [bis -3](#) [bis -5](#) [mehr als 5](#)
    - up to 3 years ago
    - more than 5 years ago
- Radiologie-Befunde (Rados via Miraplus) — radiology findings
  - o [aktuellster](#) / Jahr(e): [bis -0.5](#) [bis -1](#) [bis -2](#) [bis -3](#) [bis -5](#) [mehr als 5](#)
  - o Liste (Jahre): [bis -0.5](#) [bis -1](#) [bis -2](#) [bis -3](#) [bis -5](#) [mehr als 5](#)
- Mikrobiologische-Befunde (Micro via Miraplus) — microbiology findings
  - o [aktuellster](#) / Jahr(e): [bis -0.5](#) [bis -1](#) [bis -2](#) [bis -3](#) [bis -5](#) [mehr als 5](#)

**Figure 5.** Menu of Webservices for doctors letters and findings.

These are:

- Medatec Structure Definition Language (MSDL)
  - o Including the corresponding interpreter and compiler
  - o Definitions kept in text files with file extension 'msd'
- Type Programming Language (TPL)
  - o Including the corresponding interpreter
  - o Definitions kept in text files with file extension 'tpl'
- TeX-file enhancing Language (TXF)
  - o Including the corresponding interpreter
  - o Definitions kept in text files with file extension 'txf'

In principle, this approach can be considered as an early precursor to Document Type Description (DTD) and Extensible Markup Language (XML) system.

The Interactive Mask Generator (IMAGE) is an executable binary, which was developed in parallel to the MSDL module.

With these tools, we define both the scope of functions and the program flow for a complete project.

The executable programs are merely single basic modules (kernel) that load, interpret, represent, and store statements from those definitions.

For each project, definition files are created that provide all information necessary for the project (table structures including dependencies, the contents of the input forms and procedural relationships).

These definitions are parsed by the MSDL interpreter

and then translated once via the MSDL compiler into a database structure, named structure-DB (strDB). The appearance and functionality of the system will later be determined solely by such a strDB. In a second step, the placement of the fields for every defined input form is determined once using our interactive tool (IMAGE). This tool receives all needed information from the same source (strDB) and stores the field positions in this database (Figure 6).

Because various modules, e.g. doctor's letter or tumour documentation (TuDok), can be merged, almost any requirement of an institution is relatively easy to model.

Figure 6 shows the module 'Tumour Documentation' (TuDok) as an example for a merging candidate.

All input forms are stored in the strDB as well and every single form is generated and displayed on demand from the contents of the strDB.

Due to this procedure, the content of the system is virtually unlimited and the load on the main memory is completely independent of the number of input forms, different document types or the project size. Thus, even very large projects can be realized without difficulty.

For the TeX server, introduced in 2001, two script languages/interpreters were created that combine TeX code with database content. TPL defines the basic types of used tags and TXF extend TeX files and act as a database to TeX interface. Figure 7 shows the basic scheme of the doctor's letter system.

In the supplements (see appendix II) we have attached examples of definition files without further explanation.

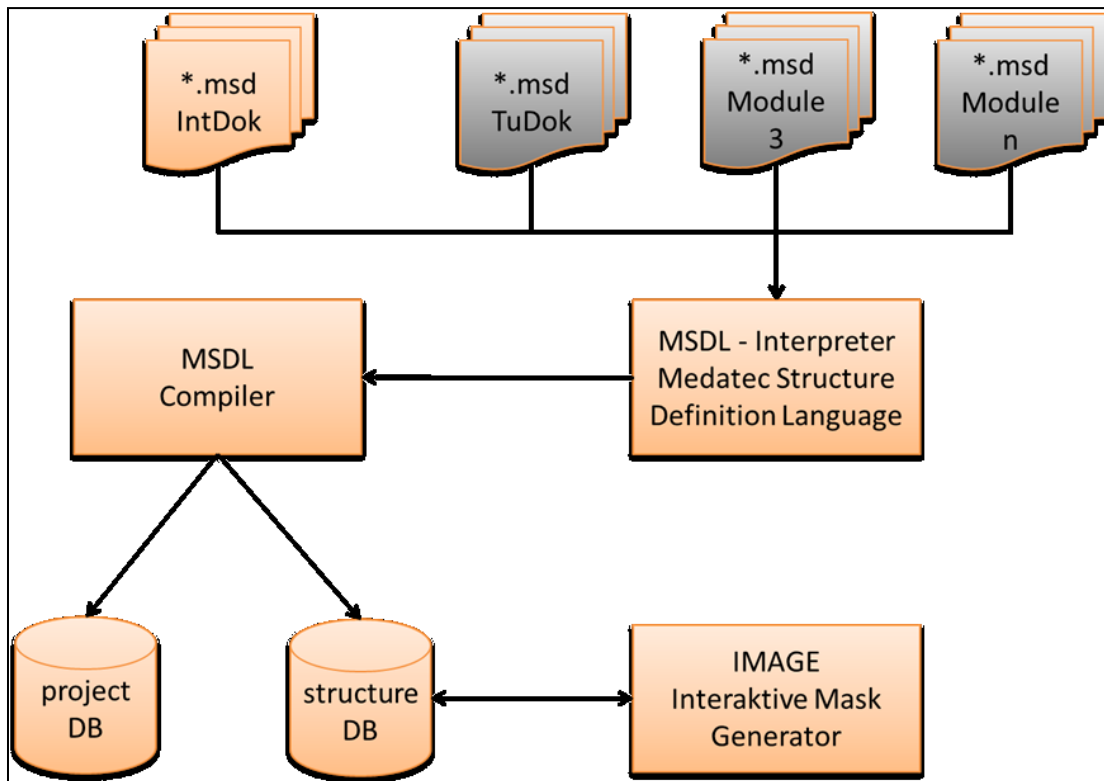


Figure 6. Schematic of the project definition process.

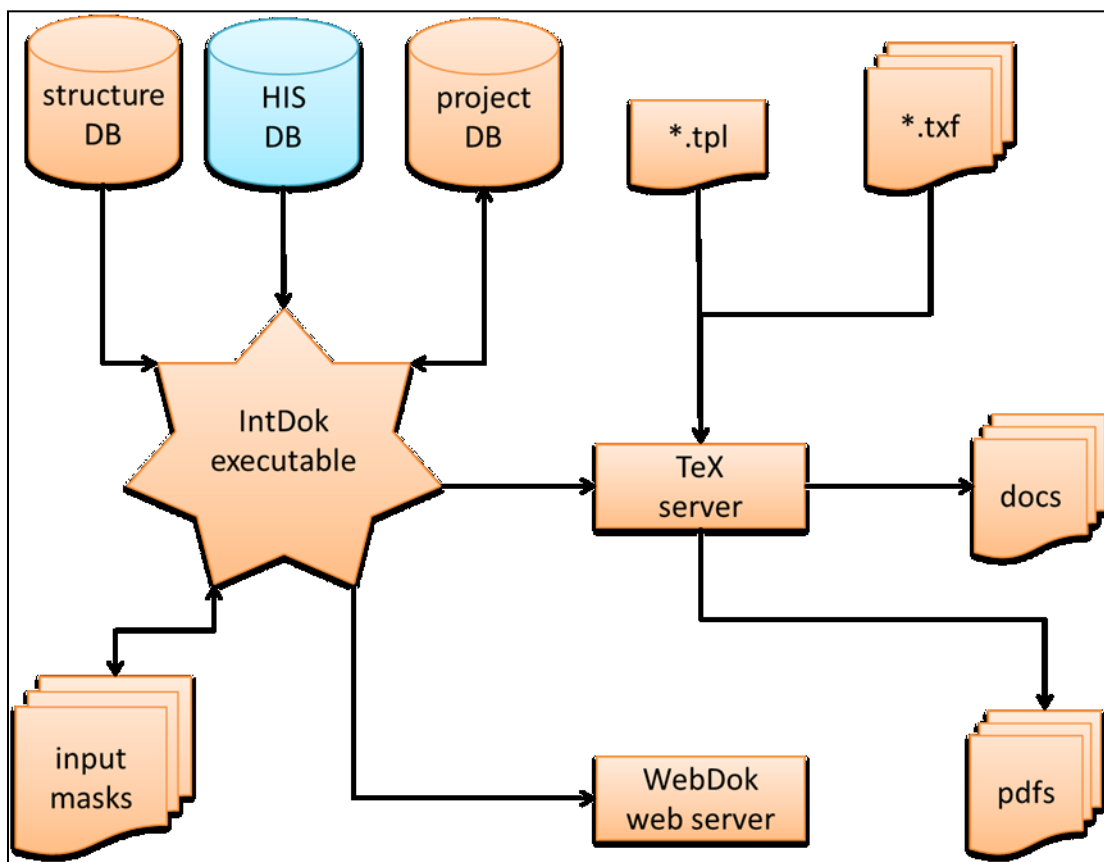


Figure 7. Internal schematic of module IntDok as a part of MEDATEC.



## Results

The problems and requirements of creating medical documents discussed in this manuscript have been effectively and smoothly resolved with the doctor's letter and finding system IntDok. The reader of those documents gets a comprehensive and quick overview of a patient's history with the help of the hierarchical order of diagnoses, histologies and therapies. Document creation is controlled with a strictly procedural approach and the user is supported by many valuable and essential tools such as reusable text blocks, extensible lists and import and export functions. HL7 interfaces to the hospital information system (HIS) and departmental information system (MOSAIQ™) provides all demographic patient data, addresses and content of those systems to the user. With templates, it is easy to implement the integration of barcodes to meet requirements set by other systems or organizations.

The software offers a sophisticated mail merge mechanism. Hierarchical text blocks of diagnoses and therapies or reporting text from former documents are easily imported into the current document and the user only has to add new diagnoses, therapies or text. Export is possible to PDF or HTML/XML format. A TeX server automatically creates document layout so that uniform appearance is guaranteed; the adaption to the corporate design of our hospital is an easy task. In addition, including a speech recognition module presents no problem. The documents are stored in a MS-SQL database, independent of any proprietary formats as, e.g., MS-Word. All documents are accessible within a web server. The database comprises all doctor's letters and reports created in our department since 1991 – approximately 230,000 documents to date. All documents can be transmitted to other systems in PDF format, retaining all necessary metadata, for example diagnoses codes as ICD and qualifiers (e.g. doctor's letter or finding).

## Discussion

The software IntDok allows for quick and structured creation of documents, enabling the handling of hierarchical blocks of diagnoses and therapies and ensures the optimum use of human resources with sophisticated tools. Within a very short amount of time, the user is able to edit very complex oncological doctor's letters. IntDok is integrated in an excellent way into our multimodal document management [12], introduced in 2013. For over four years, our department has been working almost completely without any 'analog' medium (film- and paperless) [13, 14]. Sending doctor's letters or reports to remote institutions is the last process linked to paper. We are currently negotiating with an external service provider to receive our documents electronically. The service provider will then send the documents by post to the recipients (patients, doctors in private practice and external clinics). We have simultaneously submitted a request for approval of this procedure to the state data protection authorities. If we succeed in this

step, our clinic will also be completely paperless in external communication.

Due to the modular design of the system new requirements can be implemented easily. Thus, the system can be used in virtually all medical disciplines, and it is multi-user enabled, serving, for example multiple departments within one clinic. The software provides a high level of configurability and, in addition to standard doctor's letters, clinical reports (for example about surgical procedures), medical documentation or any other documents or correspondence can be created.

For many years, MEDATEC was the largest HIS implementation at Freiburg University Hospital, with around 350 workstations in daily routine use. It was used in almost all clinics of the University Hospital and consisted of the modules for example IntDok (doctor's letter), Melody (surgical management), TuDok (tumour documentation), D-Arzt (emergency outpatient unit).

Participating clinics were dermatology, radiation oncology, head and neck, gynaecology, all surgical clinics (traumatology, orthopaedics, hand and plastic surgery, general and visceral surgery, endoscopic surgery, neurosurgery), internal medicine (endoscopy) and geriatrics. This can also be seen in the attached supplements, the entire system is highly configurable and adaptable.

However, the configuration and adaptation to new requirements requires detailed knowledge of the internal structure and, above all, of the self-developed scripting languages.

But due to the fact that the system is not commercially available, it is difficult to implement and maintain in external facilities. Moreover we cannot currently provide the necessary support to external institutions.

The software system is constantly evolving and new features continue to be added.

Finally, we are currently planning to port the software system presented here to a 64 bit platform, so that it can make use of larger amounts of memory and increased stability on 64-bit operating systems.

The current HIS systems (e.g. Orbis from Agfa, Millennium/ISH-Med from Cerner, CGM Clinical from CompuGroup, I-Soft from i-Solutions Health and MCC from Meierhofer) have a very good range of functions; however, in our opinion they are very slow in implementing adaptations or even innovative solutions. ISH-Med and SAP offer at least modular and adaptive structures, but are often classified by users as unintuitive and unwieldy.

In addition, none of these manufacturers has attempted so far to control linear accelerators (linac). In view of the relatively manageable number of radiation therapy facilities, the HIS manufacturers avoid the costly step of solving the difficult technical and legal requirements of linac control. Currently only MOSAIQ (Elekta) and ARIA (Varian) are available for radiation oncologies; Raysearch is about to launch the RAYCARE system. However, both Elekta and Varian have only started to develop towards HIS function-

ality in recent years. Therefore, external but integrative software solutions are currently still necessary (as here e.g. IntDok) if you want to work continuously and effectively in a digital environment.

## Conclusion

Creation of documents using the presented system is fast, simple and well-structured. Sophisticated tools ensure the optimal use of human resources and time. The hierarchical blocks of diagnoses, histologies and therapies give the reader a chronological and rapid overview of the previous course of therapy.

As discussed above, the entire system can be used in almost any clinical area, as it can meet all the important requirements of a clinical facility.

The modular design allows fast implementation and guarantees short time-scales necessary to implement changes or new requirements.

The limiting factor for providing the system to other facilities is due to the fact that our department cannot currently bestow support for set-up and maintenance. IntDok is an important module in our overall digital work environment [15].

## References

1. Flaig M, Graeber S, Sybrecht GW, Innere MV. Use of HL7 to integrate a HIS-subsystem: limits and possibilities 2973. MEDINFO 2001. 2001;Proceedings of the 10th World Congress on Medical Informatics. IOS Press. Part vol.1, 2001, pp. 730-4. Amsterdam, Netherlands. PubMed PMID: 2977.
2. Schadow G. Krankenhauskommunikation mit HL7: Shaker Verlag; 2000.
3. Kuo JW-Y, Kuo AM-H. Integration of Health Information Systems Using HL7: A Case Study. Studies in health technology and informatics. 2017;234:188-94. PubMed PMID: Medline:28186039.
4. Knuth DE. Computers & Typesetting: Addison-Wesley Professional USA; 1986.
5. Duckett GA. MikTeX: Questions and Answers: CreateSpace Independent Publishing Platform, USA; 2016.
6. Voß H. Einführung in LaTeX: Lehmanns Media GmbH; 2016.
7. Rodger A. Improving doctors' letters. The Medical journal of Australia. 2003;178(3):144. PubMed PMID: Medline:12558495.
8. Kirrmann S, Gainey M, Rohner F, Hall M, Bruggmoser G, Schmucker M, et al. Visualization of data in radiotherapy using web services for optimization of workflow. Radiation oncology (London, England). 2015;10:22. PubMed PMID: Medline:25601225.
9. Fehily C. SQL (Database Programming) (2015 Edition): Questing Vole Press; 2014. 648 p.
10. Molinaro A. SQL Cookbook: O'Reilly Media; 2009. 595 p.
11. Bundesministerium für Umwelt NuR Richtlinie zur Verordnung über den Schutz vor Schäden durch ionisierende Strahlen (Strahlenschutzverordnung - StrlSchV)2017.
12. Fahrner H, Kirrmann S, Rohner F, Schmucker M, Hall M, Heinemann F. [Multimodal document management in radiotherapy] Multimodales Dokumentenmanagement in der Strahlentherapie. Strahlentherapie und Onkologie : Organ der Deutschen Röntgengesellschaft [et al]. 2013;189(12):1032-9. PubMed PMID: Medline:24196281..
13. Nitrosi A, Borasi G, Nicoli F, Modigliani G, Botti A, Bertolini M, et al. A filmless radiology department in a full digital regional hospital: quantitative evaluation of the increased quality and efficiency. Journal of digital imaging. 2007;20(2):140-8. PubMed PMID: Medline:17318704.
14. Nusbaum NJ. The electronic medical record and Patient-centered care. Online J Public Health Inform. 2011;3(2). PubMed PMID: Medline:23569603.
15. Heinemann F, Rohner F, Schmucker M, Bruggmoser G, Henne K, Grosu A-L, et al. [Department and patient management in radiotherapy. The Freiburg model]. Abteilungs- und Patientenmanagement in der Strahlenheilkunde. Das Freiburger Modell. Strahlentherapie und Onkologie : Organ der Deutschen Röntgengesellschaft [et al]. 2009;185(3):143-54. PubMed PMID: Medline:19330290

## Appendix

Table 2

### List of abbreviations

BDT	official standard for exchange treatment data
C++	C programming language IDE from Borland
DRAGON	Nuance Communications, Inc., Burlington, USA
DTD	Document Type Description
DV	Data processing
ELEKTA	Vendor of accelerators, radiotherapy software, R&V-System
ErgoShell	pseudo-graphical User-Interface based on MS-DOS
Harbour	xbase compiler open source
HIS	Hospital Information System

HL7	Health Level Seven (standard for exchanging information between medical applications)
HTML	Hyper Text Markup Language
ICD	International classification of disease
IMAGE	Interactive Mask Generator
IntDok	Doctor's letter and finding system (document producer and handler)
Javascript	Web scripting language from Netscape
JQuery	JavaScript library
LaTeX	Extension of TeX
MEDATEC	Medical Data and Text processing with Computer
MikTeX	Extension of TeX
MOSAIQ	Departmental system (ELEKTA), R&V-System
MRT	Magnetic Resonance Tomography
MSDL	Medatec Structure Definition Language
MS-SQL	Data-management System (Microsoft)
MS-DOS	Microsoft Disk Operating System
MS-Word	text processing system
OpenOffice	MS-Office analog package
PC	Personal computer
PDF	Portable Document Format
PHP	Web scripting language open source
PID	patient identifier
SQL	structured query language
strDB	MEDATEC structure database
TCP	Transmission Control Protocol
TeX	Typesetting system
TIFF	Tagged Image File Format
TPL	Type Programming Language
TXF	TeX-file enhancing Language
Windows	Operationg system (Microsoft)
Windows 3.1	Operationg System Microsoft
WordPerfect	old text processing system
WordStar	old text processing system
XML	Extensible Markup Language

## Appendix II

The following supplements would comprise 89 pages. Therefore we have added only a small selection; these are original project files in German language and formatted in Courier New for readability.

\*\*\*\*\*  
Supplements  
\*\*\*\*\*

File: TYPES.TPL

-----

```

*=====
* TYPES.TPL - MEDATEC - Type Programming Language
*
* R. Preiß / F. Heinemann / M. Vogel (c) 1991-2018
*-----
* B A S I S - Typ Definitionen
*-----

// Allgemeinchirurgischer Brief
BRFALLG ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// Allgemeinchirurgischer Kinderbrief
BRFALLK ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// Allgemeinchirurgischer Endoskopiebericht
BRFENDO ::= [ AllgChirEndoData, Adressaten, Kopien, InfoCopy ]
// Allgemeinchirurgischer OPBericht
OPALLG ::= [ AllgChirOPData, Adressaten, Kopien, InfoCopy ]
// Allgemeinchirurgischer OPBericht-Kinder
OPALLK ::= [ AllgChirOPData, Adressaten, Kopien, InfoCopy ]

// Orthopädischer Brief
BRFORT ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// Orthopädischer Brief Reichelt
BRFOR1 ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// Orthopädischer Ambulanzbrief
BRFOR2 ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// Orthopädischer OPBericht
OPORT ::= [ AllgChirOPData, Adressaten, Kopien, InfoCopy ]

// Unfallchirurgischer Brief
BRFUNF ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// Unfallchirurgischer Kurzbrief
KRFUNF ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// Unfallchirurgischer OPBericht
OPUNF ::= [ AllgChirOPData, Adressaten, Kopien, InfoCopy ]

// Handchirurgischer Brief
BRFHAND ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// dito mit 3 Unterschriften
BRFHAN3 ::= [ AllgChirBriefData, Adressaten, Kopien, InfoCopy ]
// Handchirurgischer OPBericht
OPHAND ::= [ AllgChirOPData, Adressaten, Kopien, InfoCopy ]

BRFSTTHA ::= [ StthBriefData, Adressaten, Kopien, InfoCopy ]

```

```

KLISTTHA ::= [ StthBriefData, Adressaten, Kopien, InfoCopy ]
MVZSTTHA ::= [ StthBriefData, Adressaten, Kopien, InfoCopy ]
KLISTTHP ::= [ StthBriefData, Adressaten, Kopien, InfoCopy ]

StthBriefData ::= [ PatBrief, Diktanden, Bearbeiter, BriefDatum,
                   Aufenthalt, FallNummer, RecordID, GrussFormel,
                   HsetDiagTher, NebenDiag, BriefText,
                   EndeFormel, Nachrichtlich ]

AllgChirBriefData ::= [ PatBrief, Diktanden, Bearbeiter, BriefDatum,
                       Aufenthalt, GrussFormel,
                       Diagnosen, Histologien, Therapien, NebenDiag,
                       BriefText, EndeFormel, Nachrichtlich ]

AllgChirOPData    ::= [ PatOP, Diktanden, Operateure, Anaesthesisten,
                       BriefDatum, Aufenthalt, OPSchwestern, Bearbeiter,
                       OPDatum, Diagnosen, Therapien, BriefText, Nachrichtlich
                       ]

AllgChirEndoData ::= [ PatOP, Diktanden, Untersucher, Pflegekraefte,
                       BriefDatum, Aufenthalt, UndersDatum, UndersZeitVon,
                       UndersZeitBis, BriefText, EndeFormel, Praemedikation,
                       Geraete, UMethod, TMethod, Nachrichtlich ]

Untersucher      ::= STRING
Pflegekraefte   ::= STRING
UndersDatum      ::= STRING
UndersZeitVon    ::= STRING
UndersZeitBis   ::= STRING
Praemedikation  ::= STRING
Geraete          ::= STRING
UMethode         ::= STRING
TMethod          ::= STRING

PatBrief ::= [ Name, Vorname, Titel, GebDatum, Geschlecht, Wohnort, Strasse,
              PatPiz ]
PatOP    ::= [Name, Vorname, Titel, GebDatum, Geschlecht, Wohnort, Strasse,
              Kasse, Station]

Name      ::= STRING
Vorname   ::= STRING
GebDatum  ::= STRING
Wohnort   ::= STRING
Strasse   ::= STRING
Kasse     ::= STRING
Station   ::= STRING
PatPiz    ::= STRING

Adressaten ::= MegaAdressat*
MegaAdressat ::= [Adressat, Intro, KopieFirst]

Adressat ::= [Anrede, Bezeich, Geschlecht, Institut, Abteilung,
              FunktionsEinheit, Titel, AdrName, Vorname, AdrStrasse,
              AdrWohnort]

```

```
Anrede           ::= STRING
Bezeich          ::= STRING
Geschlecht       ::= STRING
Institut         ::= STRING
Abteilung        ::= STRING
FunktionsEinheit ::= STRING
Titel            ::= STRING
Intro            ::= STRING
KopieFirst       ::= STRING
AdrName          ::= STRING
AdrWohnort       ::= STRING
AdrStrasse       ::= STRING
```

```
Kopien ::= KopienBezeichner*
KopienBezeichner ::= STRING
```

```
InfoCopy ::= [InfoCopyText]
InfoCopyText ::= STRING
```

```
Diktanden ::= Diktand | ZweiDiktanden | DreiDiktanden
```

```
Diktand ::= [ Kuerzel, Name, Titel, Funktion ]
  Funktion ::= STRING
  Kuerzel ::= STRING
ZweiDiktanden ::= [ Diktand1, Diktand2 ]
  Diktand1 ::= [ Diktand ]
  Diktand2 ::= [ Diktand ]
DreiDiktanden ::= [ Diktand1, Diktand2, Diktand3 ]
  Diktand3 ::= [ Diktand ]
```

```
Operateure ::= Operateur | ZweiOperateure | DreiOperateure | VierOperateure
```

```
Operateur ::= [ Kuerzel, Name, Titel, Funktion ]
ZweiOperateure ::= [ Operateur1, Operateur2 ]
  Operateur1 ::= [ Operateur ]
  Operateur2 ::= [ Operateur ]
DreiOperateure ::= [ Operateur1, Operateur2, Operateur3 ]
  Operateur3 ::= [ Operateur ]
VierOperateure ::= [ Operateur1, Operateur2, Operateur3, Operateur4 ]
  Operateur4 ::= [ Operateur ]
```

```
Anaesthesisten ::= Anaesthesist | ZweiAnaesthesisten
```

```
Anaesthesist ::= [ Kuerzel, Name, Titel, Funktion ]
ZweiAnaesthesisten ::= [ Anaesthesist1, Anaesthesist2 ]
  Anaesthesist1 ::= [ Anaesthesist ]
  Anaesthesist2 ::= [ Anaesthesist ]
```

```
OPSchwestern ::= STRING
Bearbeiter    ::= STRING
BriefDatum    ::= STRING
Aufenthalt    ::= STRING
FallNummer    ::= STRING
RecordID      ::= STRING
GrussFormel   ::= STRING
```

```
EndeFormel ::= STRING
BriefText  ::= STRING
OPDatum    ::= STRING

Diagnosen ::= Diagnose*
Diagnose   ::= [ DiagnoseText, ICD ]
  DiagnoseText ::= STRING
  ICD          ::= STRING

Histologien ::= Histologie*
Histologie  ::= [ HistologieText, KEY ]
  HistologieText ::= STRING
  KEY            ::= STRING

Therapien  ::= Therapie*
Therapie   ::= [ TherapieText, IKPM ]
  TherapieText ::= STRING
  IKPM         ::= STRING

NebenDiag ::= [NebenDiagText]
NebenDiagText ::= STRING

HsetDiagTher ::= HsetDHT*

HsetDHT      ::= [HsetDiagText, HsetDiagIcd, HsetHistL, HsetTherL]

HsetHistL    ::= HsetHist*
HsetHist     ::= [HsetHistText, HsetHistKey]

HsetTherL    ::= HsetTher*
HsetTher     ::= [HsetTherText, HsetTherKey]

HsetDiagText ::= STRING
HsetDiagIcd  ::= STRING
HsetHistText ::= STRING
HsetHistKey  ::= STRING
HsetTherText ::= STRING
HsetTherKey  ::= STRING

Nachrichtlich ::= [ NachrErsteZeile, NachrFolgeZeilen ]
NachrErsteZeile ::= STRING
NachrFolgeZeilen ::= NachrFolgeZeile*
NachrFolgeZeile ::= STRING
```

---

File: MSDL.MSD

-----  
#include "mini.msdl"

```

=====
* MSDL.MSD - M-edatec S-structure D-efinition L-anguage *
* R. Preiß / F. Heinemann / M. Vogel (c) 1991-2018
*-----
* Definition der Steuertabelleen *
* XBROWSE XBrowser-Steuertabelle
* RELATION Relationen-Steuertabelle
* BOXMENU Boxmenue-Steuertabelle
* BOXPAGES BoxPages-Steuertabelle
* PROJEKT Projekt-Steuertabelle
* RIGHTS Rechte [unparam.]-Steuertabelle
* DOCASE CaseKonstrukt-Steuertabelle
* LEITST Leitstellen-Steuertabelle
* SICHTEN Leitstellen-Sichten-Steuertabelle
* ZUSTAND Zustand-Steuertabelle
* AKTIONEN Aktionen-Steuertabelle
* ETIDEF Etiketten-Steuertabelle
* REPDEF Report-Steuertabelle
* SET SET-Steuertabelle
* QUERV Querverweis-Steuertabelle
* DOKART Dokumentations-Art-Steuertabelle
* DOKUS Dokumentationen-Steuertabelle
* MERGE Merge-Steuertabelle
* MRG_VAR Merge-Variablen
* MEM_ART Memo-Arten
* STD_GRP Standard-Textblockgruppen
* DBBAUM Datenbank-Hierarchie
* ESET Eingabe-Sets
* HSET Hierarchische Eingabe-Sets
* IMPORTCMD Importaufträge
* IMPORT DB-Importabläufe
* JOBART Jobart-Definitionen
* MSDFUNC Eigene Funktionen
* CHECKSET Set für Plausichecks für DB-Einträge
* CHECKVAL Check-Definitionen, die in CHECK-SET verwendet werden*
*-----
DEFINE DORG_DB
*-----
* Neuer TBROWSER
*-----
:XBROWSE
INFO (XBrowser-Steuertabelle)
FILE (XMSD_XBR.DBF)
ART (S)
DEFINE INDEX
: DB_NAME (XBROWSE) FILE (XMSD_XBR.NTX) INDEX (YMSDKEY)
END
DEFINE MSD_STRU
:MSDID NAME (YMSDID ) TYPE (ID)
:MSDPAR NAME (YMSDPAR) TYPE (ID) && ID der zugehörigen Datenbank
VAR (BRO_DB)
:MSDKEY NAME (YMSDKEY) TYPE (C) && Browser-Name
LEN (30)
:INDEX NAME (YINDEX ) TYPE (C) && ID der zugehörigen Indizes
LEN (54) VAR (BRO_INDEX)
:BROSPALT NAME (YBROSPALT) TYPE (C) && Nummer der zugeh. Spalten
LEN (54) VAR (BRO_SPALT)

```



```

:DEFSPALT  NAME(YDEFSPALT) TYPE(C)  && Spalte, in die zu Beginn
           LEN(2) VAR(DEF_SPALT)    && gesprungen wird
:LIST      NAME(YLIST ) TYPE(C)     && Browser-Anzeige
           LEN(254) VAR(BRO_LIST )
:COLOR     NAME(YCOLOR)  TYPE(C)    && Browser-Farben
           LEN(300) VAR(BRO_COL)
:TITEL     NAME(YTITEL ) TYPE(C)    && Browser-Titel
           LEN(40) VAR(BRO_TITEL)
:SPTITEL   NAME(YSPTITEL ) TYPE(C)  && BrowserSpalten-Titel
           LEN(254) VAR(BRO_SPTITEL)
:GRENZEN   NAME(YGRENZ ) TYPE(C)    && Grenzen
           LEN(300) VAR(BRO_GRENZ)
:KEY       NAME(YKEY ) TYPE(C)      && Anwählbare Tasten
           LEN(50) VAR(BRO_KEY )
:KEYMSG    NAME(YKEYMSG) TYPE(C)    && Tasten-Erklärung
           LEN(110) VAR(BRO_MSG )
:FUNC      NAME(YFUNC ) TYPE(C)     && Name der selbstdefinierten
           LEN(10) VAR(SEL_FUNC)    && Funktion.
:MASKE     NAME(YMASKE ) TYPE(C)    && Name der zugehörigen Maske
           LEN(10) VAR(BRO_MSK)
:INS_FUNC  NAME(YINSFNC) TYPE(C)    && 'Einfüge'-Funktion
           LEN(10) VAR(INS_FUNC)
:KOR_FUNC  NAME(YKORFNC) TYPE(C)    && 'Korrektur'-Funktion
           LEN(10) VAR(KOR_FUNC)
:DEL_FUNC  NAME(YDELFC) TYPE(C)     && 'Löschen'-Funktion
           LEN(10) VAR(DEL_FUNC)
:BRO_FUNC  NAME(YBROFNC) TYPE(C)    && Func. Umschaltung andere Daten-
           LEN(10) VAR(BRO_FUNC)    && Spalte
:USR_FUNC  NAME(YUSRFNC) TYPE(C)    && Spez. 'User'-Funktion
           LEN(10) VAR(USR_FUNC)
:ZEILEN    NAME(YZEILEN) TYPE(N)    && Fenster-Höhe
           LEN(2) VAR(ZEILEN)
:SPALTEN   NAME(YSPALTEN) TYPE(N)   && Fenster-Breite
           LEN(2) VAR(SPALTEN)
:EBREITE   NAME(YEBREITE) TYPE(N)   && Eingabe-Feld-Breite
           LEN(2) VAR(EBREITE)
:SBREITE   NAME(YSBREITE) TYPE(N)   && Browser-Breite
           LEN(2) VAR(SBREITE)
:ERGEBNIS  NAME(YERGEBNIS) TYPE(C)  && Ergebnis-Ausdruck
           LEN(100) VAR(ERGEBNIS)
:SEEK_CMD  NAME(YSEEK_CMD) TYPE(C)  && Seek-Befehl
           LEN(500) VAR(SEEK_CMD)
:FREEZE    NAME(YFREEZE) TYPE(N)    && Eingefrorene Spaltenz. von li
           LEN(3) VAR(FREEZE)
:SECWIN    NAME(YSECWIN) TYPE(N)    && Zweites Fenster-Breite
           LEN(2) VAR(SECWIN)
:SWIDESCR  NAME(YSWIDESCR) TYPE(C)  && Zweites Fenster: feste Beschr.
           LEN(200) VAR(SWIDESCR)
:SWINHALT  NAME(YSWINHALT) TYPE(C)  && 2. Fenster: Ausdrücke
           LEN(500) VAR(SWINHALT)
:SWINEXCP  NAME(YSWINEXCP) TYPE(C)  && 2.Fenster: Ausnahme-Ausdrücke
           LEN(500) VAR(SWINEXCP)  && z.B. bei 2 Ausdrücke in 1 Zeile
:EING_DEF   NAME(YEINGDEF) TYPE(C)   && Vordef. Eingabe/Spalte
           LEN(100) VAR(EINGDEF)
:EING_SEEK  NAME(YEINGSEEK) TYPE(C)  && Vordef. Seek-Command/Spalte
           LEN(100) VAR(EINGSEEK)
:CARGO     NAME(YCARGO) TYPE(C)     && frei verwendbare Infos

```

```

        LEN(100) VAR(CARGO)
:HILFE      NAME(YHELPTABL) TYPE(C) && Hilfetafel zum Selektor
        LEN(8) VAR(BRO_HELP)
END

*-----
*  Relationen
*-----
:RELATION
INFO(Relationen-Steuertabelle)
FILE(XMSD_REL.DBF)
ART(S)

DEFINE MSD_STRU
  :MSDID      NAME(YMSDID ) TYPE(ID)
  :MSDPAR     NAME(YMSDPAR) TYPE(ID)  && ID der Parent-Datenbank
             VAR(REL_DB)
  :MSDKEY     NAME(YMSDKEY) TYPE(C)   && MSD-Name
             LEN(30)
  :INTO_DB    NAME(YINTODB) TYPE(C)   && ID der abhängigen Datenbank
             LEN(10) VAR(REL_CHILD)
  :REL_KEY    NAME(YRELKEY) TYPE(C)   && Relationskey für Indexsuche
             LEN(30) VAR(REL_KEY)
  :VIA_IND    NAME(YVIAIND) TYPE(C)   && Index der abhängigen DB
             LEN(10) VAR(REL_VIANTX) && für diese Suche
  :GET_LAST   NAME(YGETLST) TYPE(C)   && "J", wenn Relation auf letzten
             LEN(1)  VAR(REL_GETLST) && Eintrag einer ParentID
END

DEFINE INDEX
  : DB_NAME(RELATION) FILE(XMSD_REL.NTX) INDEX(YMSDKEY)
END

*-----
*  Neues MENUE
*-----
:BOXMENUE
INFO(Boxmenue-Steuertabelle)
FILE(XMSD_BXM.DBF)
ART(S)

DEFINE MSD_STRU
  :MSDID      NAME(YMSDID ) TYPE(ID)
  :MSDKEY     NAME(YVARIABLE) TYPE(C) LEN(10)  && Variablen-Name
  :PAGES      NAME(YPAGES ) TYPE(C) LEN(100) VAR(BXM_PAGES)
  :ESC_PROC   NAME(YESCPROC ) TYPE(C) LEN(20)  VAR(BXM_ESCPR)
END

DEFINE INDEX
  : DB_NAME(BOXMENUE) FILE(XMSD_BXM.NTX) INDEX(YVARIABLE)
END

*-----
*  Neues Boxmenue Pages
*-----
:BOXPAGES
INFO(BoxPages-Steuertabelle)
FILE(XMSD_BXP.DBF)
ART(S)

```

```

DEFINE INDEX
  : DB_NAME (BOXPAGES) FILE (XMSD_BXP.NTX) INDEX (YVARIABLE)
END
DEFINE MSD_STRU
  :MSDID      NAME (YMSDID )      TYPE (ID)
  :MSDKEY     NAME (YVARIABLE)    TYPE (ID)  && ID der zugehörigen Datenbank
  :PAGETITEL  NAME (YPAGETITEL)  TYPE (C)  LEN ( 40) VAR (BXP_PGTIT)
  :MODULES    NAME (YMODULES)     TYPE (C)  LEN (300) VAR (BXP_MOD)
  :MODFUNC    NAME (YMODFUNC)     TYPE (C)  LEN (300) VAR (BXP_FNC)
END

*-----
*  PROJEKT-Definitionen
*-----
:PROJEKT
INFO (Projekt-Steuertabelle)
FILE (XMSD_PRO.DBF)
ART (S)
DEFINE MSD_STRU
  :MSDID      NAME (YMSDID )      TYPE (ID)
  :MSDKEY     NAME (YVARIABLE)    TYPE (C)  LEN (10)  && Variablen-Name
  :ROOT       NAME (YROOT )      TYPE (C)  LEN (50)  VAR (PROJROOT)
  :LOKAL      NAME (YLOKAL )     TYPE (C)  LEN ( 1)  VAR (PROJLOKAL)
  :KUERZEL    NAME (YKUERZEL )   TYPE (C)  LEN (10)  VAR (PROJKURZ)
  :BESCHRB    NAME (YBESCHRB )   TYPE (C)  LEN (30)  VAR (PROJDESC)
END
DEFINE INDEX
  : DB_NAME (PROJEKT) FILE (XMSD_PRO.NTX) INDEX (YVARIABLE)
END

*-----
*  unparam. RECHTE-Definitionen
*-----
:RIGHTS
INFO (Rechte [unparam.]-Steuertabelle)
FILE (XMSD_RIG.DBF)
ART (S)
DEFINE MSD_STRU
  :MSDID      NAME (YMSDID )      TYPE (ID)
  :MSDKEY     NAME (YMSDKEY)     TYPE (C)  LEN (10)  && MSD-Variablen-Name
  :VAR        NAME (YVAR )       TYPE (C)  LEN (10)  && Public-Variablen-Name
  :DESCR      NAME (YDESCR )     TYPE (C)  LEN (50)  && Beschreibung des Rechts
  :SET        NAME (YSET )       TYPE (C)  LEN (10)  && zug. Auswahlset
  :MODUL      NAME (YMODUL )     TYPE (C)  LEN (10)  && zug. Modul o. allgemein
  :MAX        NAME (YMAX )       TYPE (C)  LEN ( 1)  && enthält das max. Recht
  :NOR        NAME (YNOR )       TYPE (C)  LEN ( 1)  && enthält Default-Recht
END
DEFINE INDEX
  : DB_NAME (RIGHTS) FILE (XMSD_RIG.NTX) INDEX (YMODUL)
END

*=====
*  Case-Konstrukte *

```

```

*-----*
:DOCASE
INFO (CaseKonstrukt-Steuertabelle)
FILE (XMSD_CAS.DBF)
ART (S)
DEFINE INDEX
  : DB_NAME (DOCASE) FILE (XMSD_CAS.NTX) INDEX (YMSDKEY)
END
DEFINE MSD_STRU
  :MSDID      NAME (YMSDID      ) TYPE (ID)
  :MSDKEY     NAME (YMSDKEY     ) TYPE (C) LEN ( 10)  && MSD-Name
  :CASE       NAME (YCASE       ) TYPE (C) LEN (500)  && CASE-Bedingungen
              VAR (DOCASE_BED)
  :OTHERWISE  NAME (YOTHERWISE) TYPE (C) LEN ( 50)  && Otherwise-RetVal
              VAR (DOCASE_OTH)
  :RETURN     NAME (YRETURN     ) TYPE (C) LEN (100)  && 'normale' RetVals
              VAR (DOCASE_RET)
END

*=====
* Leitstellen-Definition (LS=Leitstelle)
*-----
:LEITST
INFO (Leitstellen-Steuertabelle)
FILE (XMSD_LST.DBF)
ART (S)
DEFINE INDEX
  : DB_NAME (LEITST) FILE (XMSD_LST.NTX) INDEX (YMSDKEY)
END
DEFINE MSD_STRU
  :MSDID      NAME (YMSDID      ) TYPE (ID)
              VAR (LS_ID)
  :MSDKEY     NAME (YMSDKEY     ) TYPE (C) LEN ( 10)  && MSD-Name
  :DB         NAME (YDB         ) TYPE (C) LEN ( 10)  && Datenbank
              VAR (LS_DB)
  :INDEX1     NAME (YINDEX1     ) TYPE (C) LEN ( 10)  && Index 1 (über KEYIDs)
              VAR (LS_NTX1)
  :INDEX2     NAME (YINDEX2     ) TYPE (C) LEN ( 10)  && Index 2 (über LS-Datum)
              VAR (LS_NTX2)
  :BFORM      NAME (YBFORM      ) TYPE (C) LEN ( 12)  && Berichts-Formular
              VAR (LS_BFORM)
  :VUDAT      NAME (YVUDAT      ) TYPE (C) LEN ( 10)  && Variablen-Namen des
              VAR (LS_VUDAT)  && Behandl-Datums in LS-DB
  :DESCR      NAME (YDESCR      ) TYPE (C) LEN ( 70)  && Beschreibung
              VAR (LS_DESCR)
  :ZSFELD     NAME (YZSFLD      ) TYPE (C) LEN (200)  && Zustands-DB-Felder
              VAR (LS_ZSFLD)
  :ZSDESCR    NAME (YZSDSC      ) TYPE (C) LEN (200)  && Beschreibung
              VAR (LS_ZSDSC)
  :LABT       NAME (YABT        ) TYPE (C) LEN ( 40)  && welche Abteilungen?
              VAR (LS_ABT)
  *:RECHT     NAME (YRECHT      ) TYPE (C) LEN ( 10)  && Rechtevariable
  *           VAR (LS_RECHT)
  :LKENN      NAME (YKENN       ) TYPE (C) LEN (  6)  && Leitstellenkennung
              VAR (LS_KENN)
  :LSRECHTE   NAME (YLSRECHTE   ) TYPE (C) LEN ( 50)  && Rechte für Bereiche
              VAR (LS_RECHTE)

```

```

      :LSDBFLD  NAME(YLSDBFLD  ) TYPE(C) LEN( 50)  && Datenbankfelder für
              VAR(LS_DBFLD)                                && Bereiche
END

*-----
*  Leitstellen-Sichten
*-----

:SICHTEN
INFO(Leitstellen-Sichten-Steuertabelle)
FILE(XMSD_LVI.DBF)
ART(S)
DEFINE INDEX
  : DB_NAME(SICHTEN) FILE(XMSD_LVI.NTX) INDEX(YMSDPAR+YMSDKEY)
END
DEFINE MSD_STRU
  :MSDID      NAME(YMSDID      ) TYPE(ID)
  :MSDPAR     NAME(YMSDPAR     ) TYPE(ID)          && ID der Leitstelle
  :MSDKEY     NAME(YMSDKEY     ) TYPE(C) LEN( 10)  && MSDL-Kurzname
  :DESCR     NAME(YDESCR     ) TYPE(C) LEN( 70)  && Beschreibung
              VAR(LV_DESCR)
  :INDEX     NAME(YINDEX     ) TYPE(C) LEN( 10)  && Index für Sicht
              VAR(LV_NTX)
  :FILTER     NAME(YFILTER     ) TYPE(C) LEN(200) && Filter-Bedingung
              VAR(LV_FILTER)
  :OPTSEEK   NAME(YOPTSEEK   ) TYPE(C) LEN( 30)  && Such-Bedingung
              VAR(LV_OPTSEEK )
  :OPTLIMIT  NAME(YOPTLIMIT  ) TYPE(C) LEN( 50)  && Such-Grenze
              VAR(LV_OPTLIMIT)
  :ANZEIGE   NAME(YANZEIG   ) TYPE(C) LEN(600)  && Anzeige
              VAR(LV_ANZEIG)
  :SINFO     NAME(YSINFO     ) TYPE(C) LEN(999)  && String für ESHINF-
              && Meldung
              VAR(LV_SINFO)
  *:IFUSERART NAME(YIFUSRA ) TYPE(C) LEN( 3)   && Bedingung für
              && Benutzerart
  *          VAR(LV_IFUSRA)
  :COND     NAME(YCOND     ) TYPE(C) LEN(100)  && Bedingung
              VAR(LV_COND)
END

*-----
*  Steuertabelle für Zustände der endlichen Automaten
*-----

:ZUSTAND
INFO(Zustand-Steuertabelle)
FILE(XMSD_ZUS.DBF)
ART(S)
DEFINE INDEX
  : DB_NAME(ZUSTAND) FILE(XMSD_ZUS.NTX) INDEX(YMSDPAR)
END
DEFINE MSD_STRU
  :MSDID      NAME(YMSDID ) TYPE(ID)
  :MSDPAR     NAME(YMSDPAR) TYPE(ID)
  :MSDKEY     NAME(YMSDKEY) TYPE(C) LEN(10)  && MSDL-Kurzname
              VAR(ZS_ID)
  :DESCR     NAME(YDESCR ) TYPE(C) LEN(40)  && Beschreibung
              VAR(ZS_DESCR)

```

```

:COLOR   NAME(YCOLOR ) TYPE(C) LEN( 8)  && Farbgebung
          VAR(ZS_COLOR)
:ZNR     NAME(YZNR   ) TYPE(C) LEN(10)  && Datenbankfeld
          VAR(ZS_ZNR)
:CODE    NAME(YCODE  ) TYPE(C) LEN( 2)  && Kodierung des Zustands
          VAR(ZS_CODE)
END

*-----
*  Steuertabelle für Aktionen der endlichen Automaten
*-----

:AKTIONEN
INFO(Aktionen-Steuertabelle)
FILE(XMSD_AKT.DBF)
ART(S)
DEFINE INDEX
  : DB_NAME(AKTIONEN) FILE(XMSD_AKT.NTX)
    INDEX(YMSDPAR+YMSDKEY)
    *INDEX(YMSDPAR+upper(YDOMAIN)+YMSDKEY)  RP 20.07.95
END
DEFINE MSD_STRU
:MSDID   NAME(YMSDID ) TYPE(ID)
:MSDPAR  NAME(YMSDPAR ) TYPE(ID)
:MSDKEY  NAME(YMSDKEY ) TYPE(C) LEN(10)
:DESCR   NAME(YDESCR ) TYPE(C) LEN(60)  && Beschreibung
          VAR(AC_DESCR)
:FROM    NAME(YFROM  ) TYPE(C) LEN(80)  && Quell-Zustand
          VAR(AC_FROM)
:TO      NAME(YTO    ) TYPE(C) LEN(80)  && Ziel-Zustand
          VAR(AC_TO)
:IF      NAME(YIF    ) TYPE(C) LEN(100) && Bedingung
          VAR(AC_IF)
*:IFUSERART NAME(YIFUSRA ) TYPE(C) LEN( 3) && Bedingung für
          && Benutzerart
*          VAR(AC_IFUSRA)
:COND    NAME(YCOND  ) TYPE(C) LEN(150) && Bedingung
          VAR(AC_COND)
:PRE     NAME(YPRE   ) TYPE(C) LEN(100) && Allgemeine
          && Vorbedingung
          VAR(AC_PRE)
:DOMAIN  NAME(YDOMAIN ) TYPE(C) LEN( 3)  && Definitionsbereich
          VAR(AC_DOM)
:ACTION  NAME(YACTION ) TYPE(C) LEN(40)  && Aktion
          VAR(AC_ACTION)
:PARA1   NAME(YPARA1 ) TYPE(C) LEN(100) && Parameter 1
          VAR(AC_PARA1)
:PARA2   NAME(YPARA2 ) TYPE(C) LEN(100) && Parameter 2
          VAR(AC_PARA2)
:INIT    NAME(YINIT  ) TYPE(C) LEN(20)  && Initialisierungs-
          && Funkt.
          VAR(AC_INIT ) && für globale Funktionen
:EXIT    NAME(YEXIT  ) TYPE(C) LEN(20)  && Exit-Funktion
          VAR(AC_EXIT )
END

*-----
*  Steuertabelle für Etiketten-Definition
*-----

```

```
*-----
:ETIDEF
INFO (Etiketten-Steuertabelle)
FILE (XMSD_ETI.DBF)
ART (S)
DEFINE INDEX
  :ETIDEF1 DB_NAME (ETIDEF) FILE (XMSD_ET1.NTX) INDEX (YMSDKEY)
  :ETIDEF2 DB_NAME (ETIDEF) FILE (XMSD_ET2.NTX) INDEX (YORIGDB)
END
DEFINE MSD_STRU
:MSDID      NAME (YMSDID ) TYPE (ID)
:MSDPAR     NAME (YMSDPAR ) TYPE (ID)
:MSDKEY     NAME (YMSDKEY ) TYPE (C) LEN (30)
:DESCR     NAME (YDESCR ) TYPE (C) LEN (30)    && Beschreibung
           VAR (ETI_DESCR)
:ANZEIGE    NAME (YANZEIGE) TYPE (C) LEN (500)  && Etiketten-Inhalt
           VAR (ETI_ANZ)
:DB         NAME (YDB ) TYPE (C) LEN (256)    && Quell-Datenbanken
           VAR (ETI_LDB)
:ORIGDB     NAME (YORIGDB ) TYPE (C) LEN ( 10)  && Original-Datenbank
           VAR (ETI_DB)
END
```