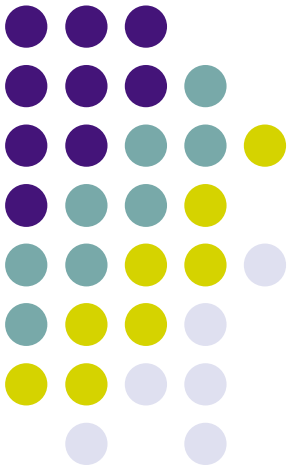
Verifying the On-Line Help System of SIEMENS Magnetic Resonance Tomographs

Carsten Sinz **Wolfgang Küchlin**

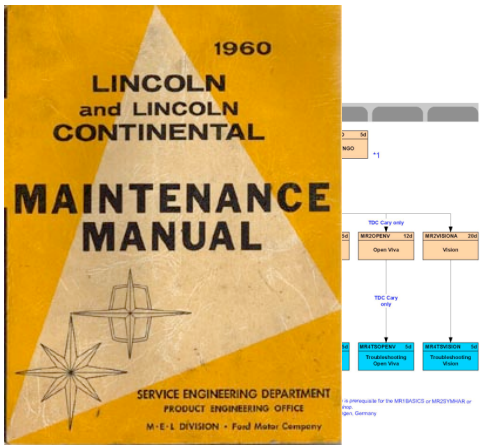
WSI for Computer Science
Symbolic Computation Group and
Steinbeis Technology Transfer Center OIT

University of Tübingen, Germany



Motivation

MR tomographs are complex products with many equipment options

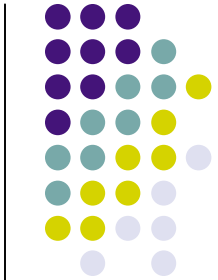


Complex documentation, favorably configured individually for each product

How to manage complexity (configurations, handbooks)?

How to avoid documentation errors?

Configurable Products



- Modular, lots of variations; adaptation to customer's needs (e.g. PCs, cars, software, telecommunication equipment, …)
- Dependencies and restrictions between components
- Configuration has been an area of AI research for many decades
  [e.g. DEC's R1/XCON System in the 80s]

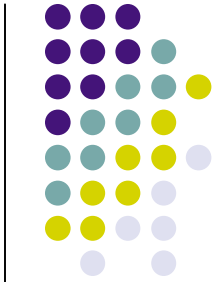
Magnetic Resonance Imaging

- Medical imaging technique
- Relies on spins of atomic nuclei
- First, MR applies strong magnetic fields to align spins in tissue
- Then, imposed RF pulses disturb spins and enforce precession of nuclei
- Coils wrapped around patient measure *relaxation* results
- Magnetic field gradients applied to determine 3D position

SIEMENS MR Tomographs: Figures and Characteristics

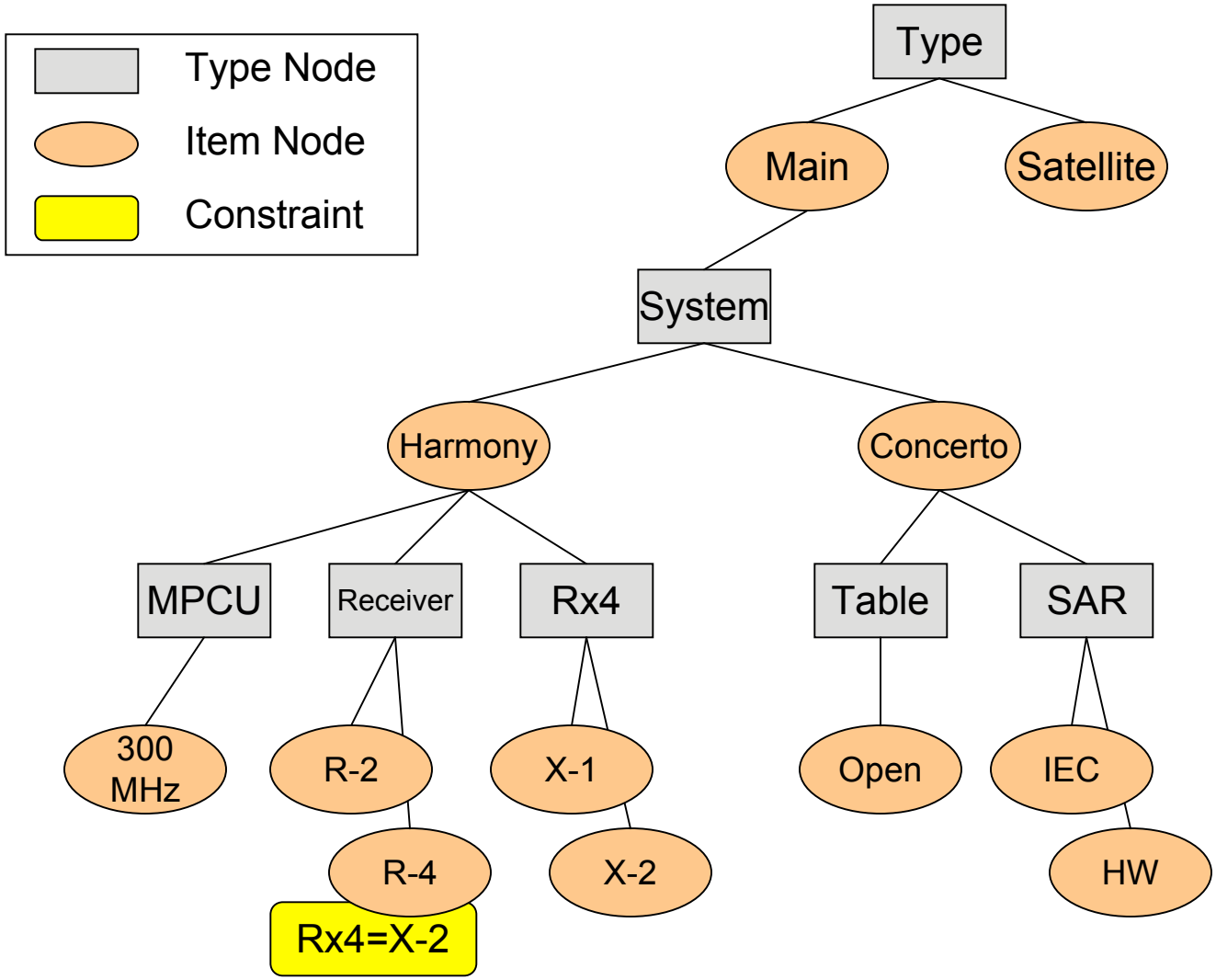
- 11 basic MR systems (model lines):
  - Concerto, Symphony, Harmony, …
- Total of 47 configurable component types, e.g.:
  - 14 different gradient power amplifiers
  - 82 different coils (for examination of different parts of the body)
  - 9 different gradient coils
  - 20 service-software add-ons
  - 2 railway mains frequency EFIs
  - 45 destination countries (e.g. India, Czech Republic,…)
  - …

SIEMENS MR Tomographs: Configuration Constraints

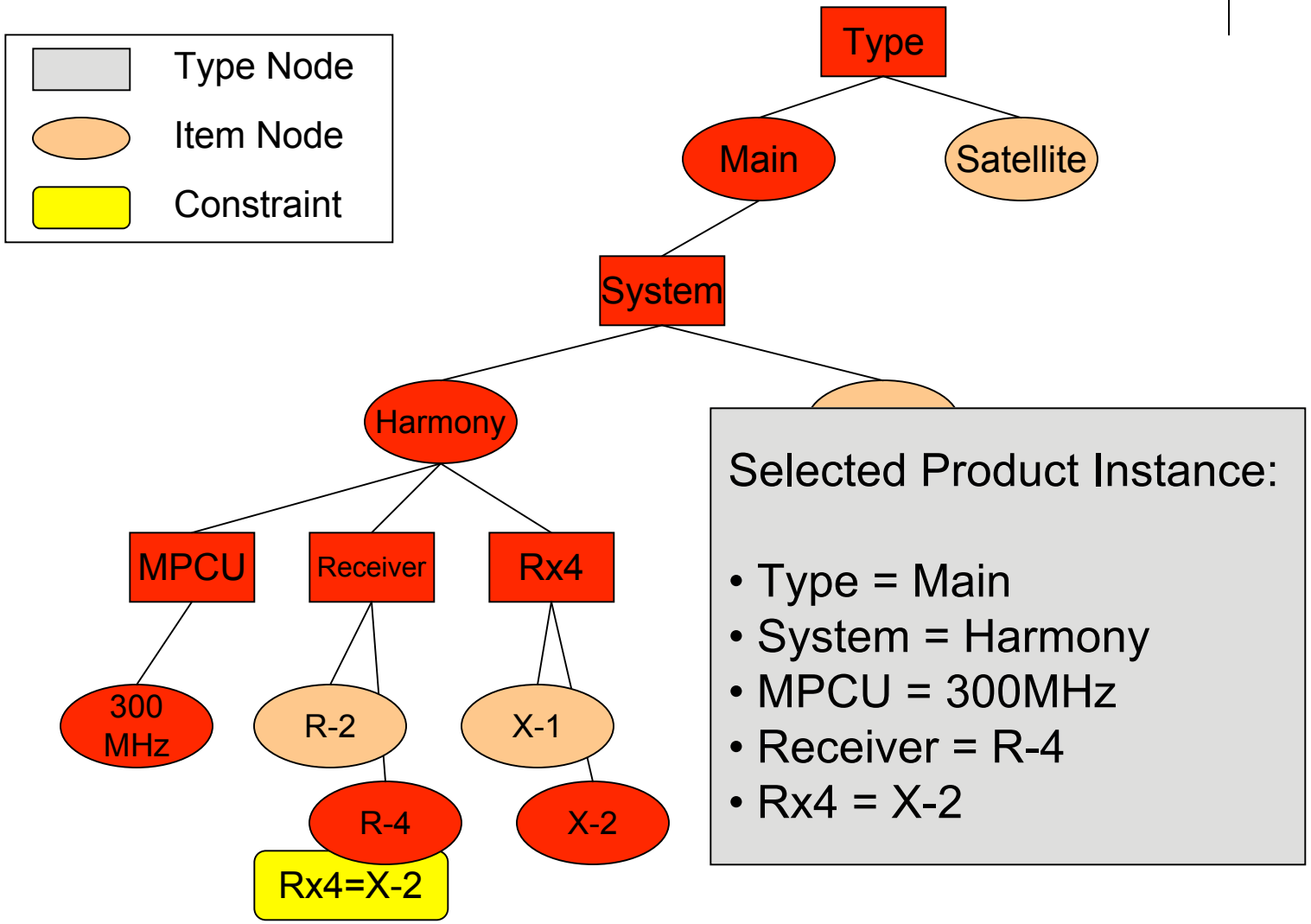


- 4 receivers require 2 receiver boards
  (RecNumOf4 requires RXNumOf2)
- Radio frequency power amplifier K2137/K2204 requires transmitter type 2
  (RFPA_K2137_K2204 requires TRA2)
- Gradient power amplifier K2209_400V requires gradient coil AS39S or AS39SR
  (GPA_K2209_400V req. GradCoil_AS39S or GradCoil_AS39SR)
- A large cabin requires body tune box 024L
  (CabinType_Large requires BTB_024L)

SIEMENS MR Tomographs: Product Structure

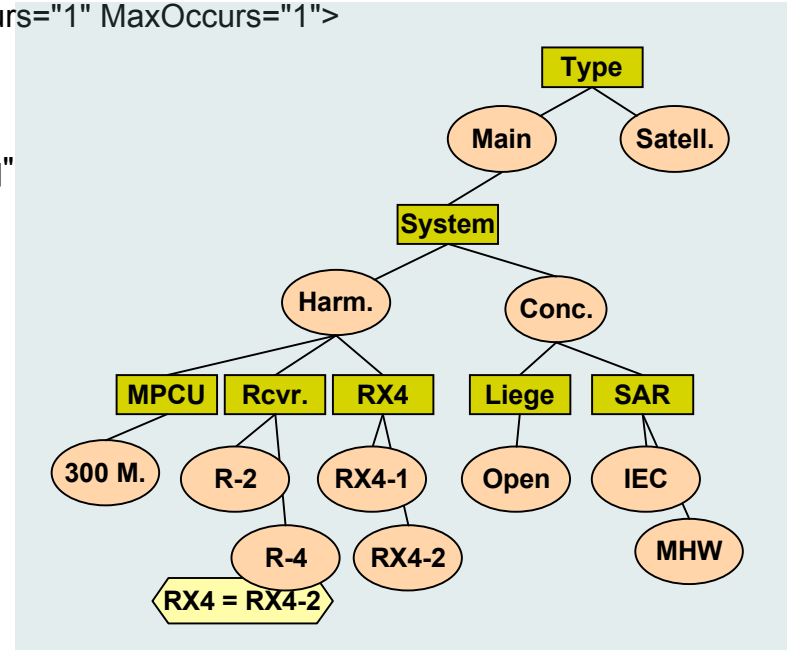


SIEMENS MR Tomographs: Product Structure

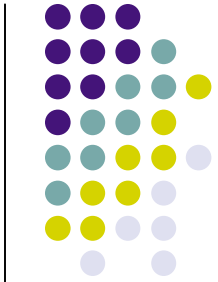


Product Structure in XML

```
<Structure>
   <Type IDREF="INT_ConsoleType" MinOccurs="1" MaxOccurs="1">
      <Item IDREF="INI_ConsoleType_Sat"/>
      <Item IDREF="INI_ConsoleType_Main">
         <SubType IDREF="INT_System" MinOccurs="1" MaxOccurs="1">
            <!-- Harmony -->
            <Item IDREF="INI_System024">
               <SubType IDREF="INT_Comp_MPCU" Default="INI_Comp_MPCU300" MinOccurs="1" MaxOccurs="1">
                  <Item IDREF="INI_Comp_MPCU300"/>
               </SubType>
               ...
               <SubType IDREF="INT_Comp_ReceiverNumOf" MinOccurs="1" MaxOccurs="1">
                  <Item IDREF="INI_Comp_ReceiverNumOf2"/>
                  <Item IDREF="INI_Comp_ReceiverNumOf4">
                     <Conditions>
                        <Condition Type="INT_Comp_RXNumOf" Op="eq"
                                   Value="INI_Comp_RXNumOf2"/>
                     </Conditions>
                  </Item>
               </SubType>
            </Item>
            ...
         </SubType>
      </Item>
   </Type>
</Structure>
```

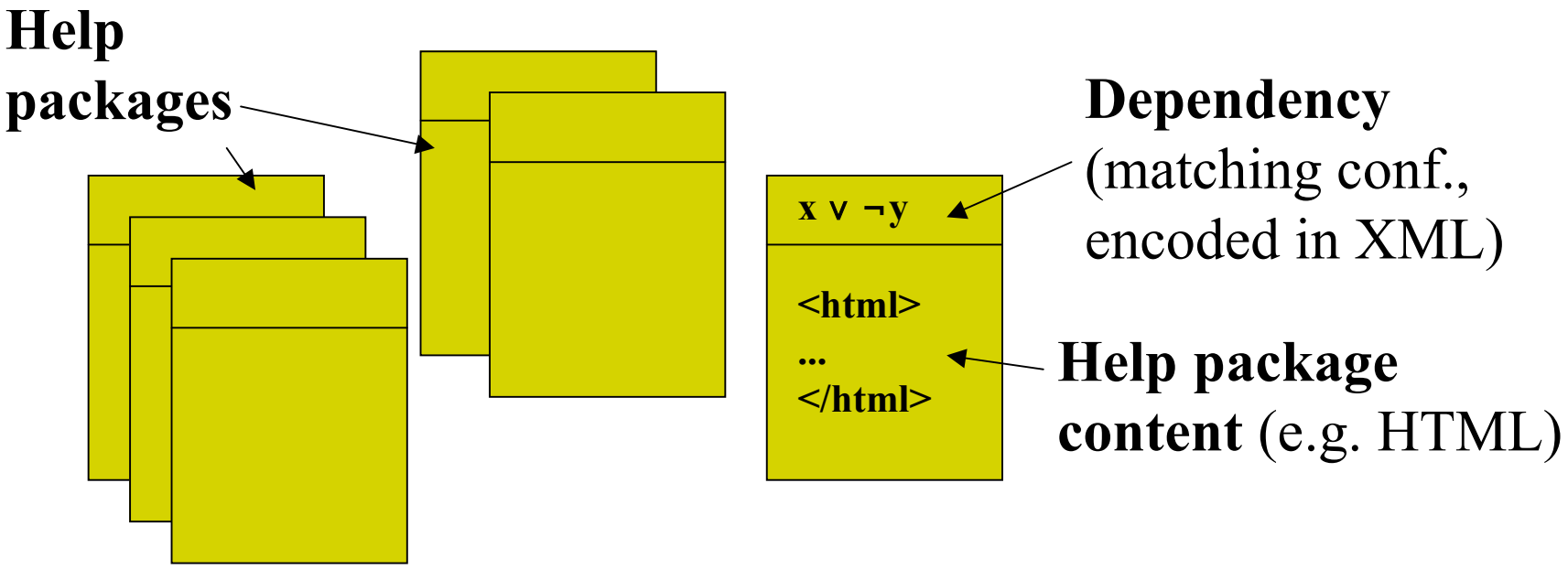


Product Manuals



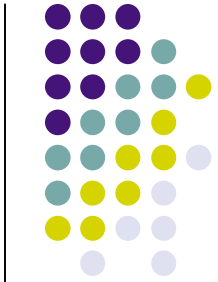
- Complex products ⇒ complex manuals
- Goal: User/service manuals reflect product instance at hand
  - Individually configured manual
  - Maintenance procedures (e.g. calibration) optimized for each product instance
- Solution: Modular help system (*Help Packages*)
  - Each help package covers only a limited aspect
  - Each help package may be configuration / workflow dependent

Modular Help System



- Automatic selection of appropriate packages to build a complete, individual documentation for each product

Consistency of Help System

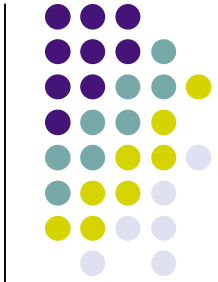


For each admissible (valid) product instance:

- No missing help packages?
- No overlaps (two or more help packages for same configuration)?

⇒ Transform to propositional logic satisfiability (SAT) problem!

Transformation to SAT



Consistency of on-line help system corresponds to validity of:

$$\text{HelpReq} \wedge \text{ValidConf} \Rightarrow \bigvee_{p \in \text{HelpPackages}} \text{HelpProv}(p)$$

$$\text{HelpReq} \wedge \text{ValidConf} \Rightarrow \neg\left(\text{HelpProv}(p_1) \wedge \text{HelpProv}(p_2)\right)$$

Transformation Detail: Valid Configurations

- Recursive definition of validity
- Validity conditions for both TYPE and ITEM nodes; e.g. for TYPE nodes:

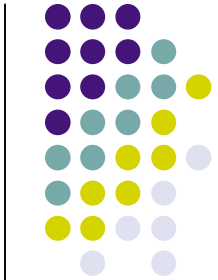
$$\mathrm{ValConfT}(t) = \mathrm{CardinalityOK}(t) \wedge \mathrm{SubItemsValid}(t) \wedge \mathrm{ForbidUnrefItems}(t)$$

$$\mathrm{CardinalityOK}(t) = \mathrm{S}_{t@\mathrm{MinOccurs}}^{t@\mathrm{MaxOccurs}}\left(\{i@\mathrm{IDREF} \mid i \in t/\mathrm{Item}\}\right)$$

$$\mathrm{SubItemsValid}(t) = \bigwedge_{i \in t/\mathrm{Item}} \left(i@\mathrm{IDREF} \Rightarrow \mathrm{ValConfI}(i)\right)$$

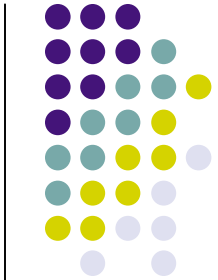
$$\mathrm{ForbidUnrefItems}(t) = \bigwedge_{i \in \mathrm{unrefItems}(t)} \neg\, i@\mathrm{IDREF}$$

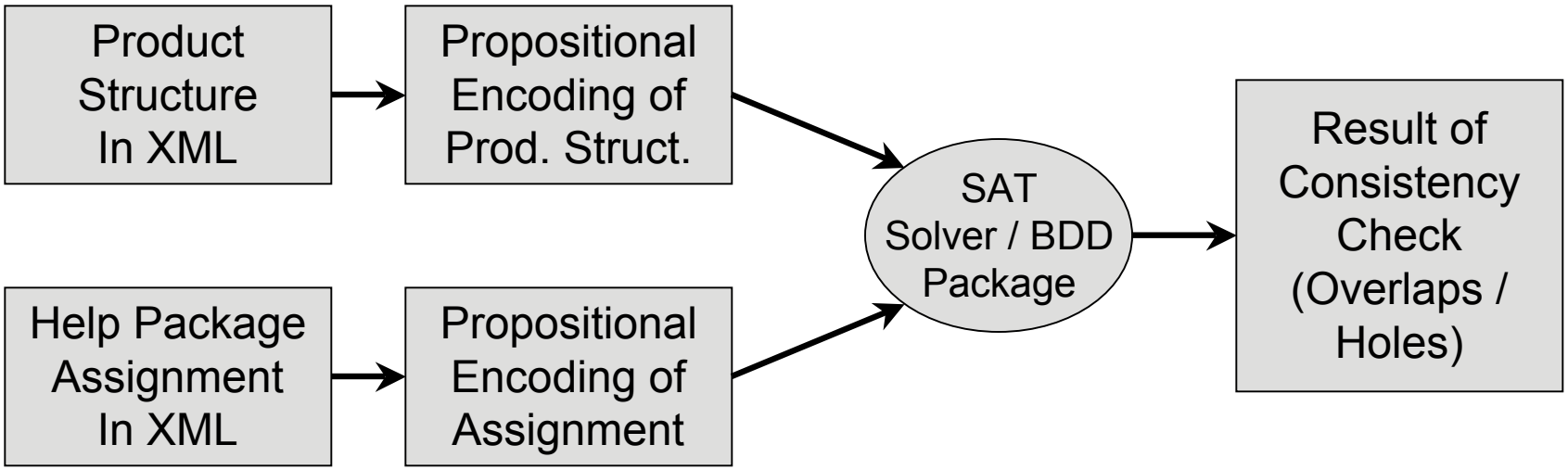
Transformation to SAT: Example



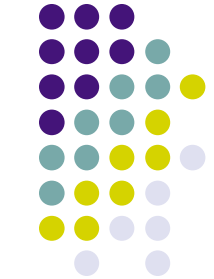
```
 1  [ValConfT(Type), SubItemsValid.Satellite]
 2
 3  (INI_ConsoleType_Sat =>
 4     (~INT_System & ~INT_Comp_MPCU & ~INT_Comp_RXNumOf &
 5      ~INT_Comp_ReceiverNumOf & ~INT_Comp_COT & ~INT_Comp_SAR)
 6  ) &
 7
 8  [ValConfT(Type), SubItemsValid.Main]
 9
10   (INI_ConsoleType_Main =>
11      (INI_System024 [Harmony] =>
12         INI_Comp_MPCU300 & ~INI_Comp_MPCU133 & ~INI_Comp_MPCU266 &
13         =1:(INI_Comp_RXNumOf1, INI_Comp_RXNumOf2) &
14         (INI_Comp_ReceiverNumOf4 => INI_Comp_RXNumOf2) &
15         =1:(INI_Comp_ReceiverNumOf2, INI_Comp_ReceiverNumOf4) &
16         ~INT_Comp_COT &
17         ~INT_Comp_SAR
18      ) &
19      (INI_System007 [Concerto] =>
...       ...
25      ) &
26      =1:(INI_System024, INI_System007)
27   ) &
28   =1:(INI_ConsoleType_Sat, INI_ConsoleType_Main) &
29
30   [Type Definitions]
31
32   (INI_ConsoleType_Sat | INI_ConsoleType_Main =>  INT_ConsoleType) & ...
```

Systematics of Consistency Checks



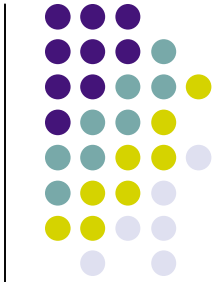


Experimental Results



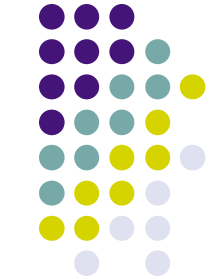
- Propositional encoding of product structure results in:
  - 1425 propositional variables
  - 11018 clauses (CNF SAT encoding)
  - 9715 formula nodes (BDD)
- Run-time for complete check:
  - 6.96 s (on 1.2 GHz Athlon with 512 MB Memory)
  - 11 model lines, 964 help contexts, 12 help packages

Technical Realization



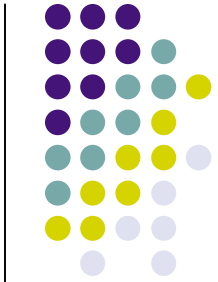
- MR product structure stored in XML data base
- SGML help pages with attached dependencies (prop. logic formulae)
- *HelpChecker* implemented in C++
  - Reads XML product structure
  - Performs checks (BDD/SAT core component)
  - Generates complete list of error cases
  - Writes results to file in XML format (missing packages / overlaps)

Deployment



- *HelpChecker* part of a larger authoring tool developed by Tanner AG, Germany for SIEMENS Medical Solutions
- Final testing phase for authoring system / *HelpChecker* in January 2005
- Deployment with new release of MR Service Software, scheduled for first half of 2005

Summary



- Modular handbooks for complex products are feasible
  - Propositional logic representation of product structure realizable
  - Automated reasoning techniques (SAT-Solvers, BDDs) sufficiently advanced
- Exact semantics (use of formal methods) enables vitally important automatic consistency checks