

Validating Syntactic and Semantic Errors of a Domain-Specific Language for Defining Clinical Pathways in Integrated Care Platforms

Selin Erdem, 27.09.2021, Munich

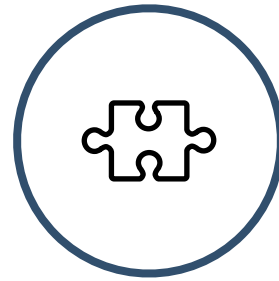
Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
www.matthes.in.tum.de



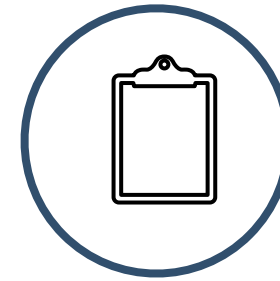
Motivation



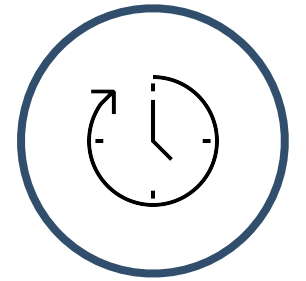
Research
Questions



Solution



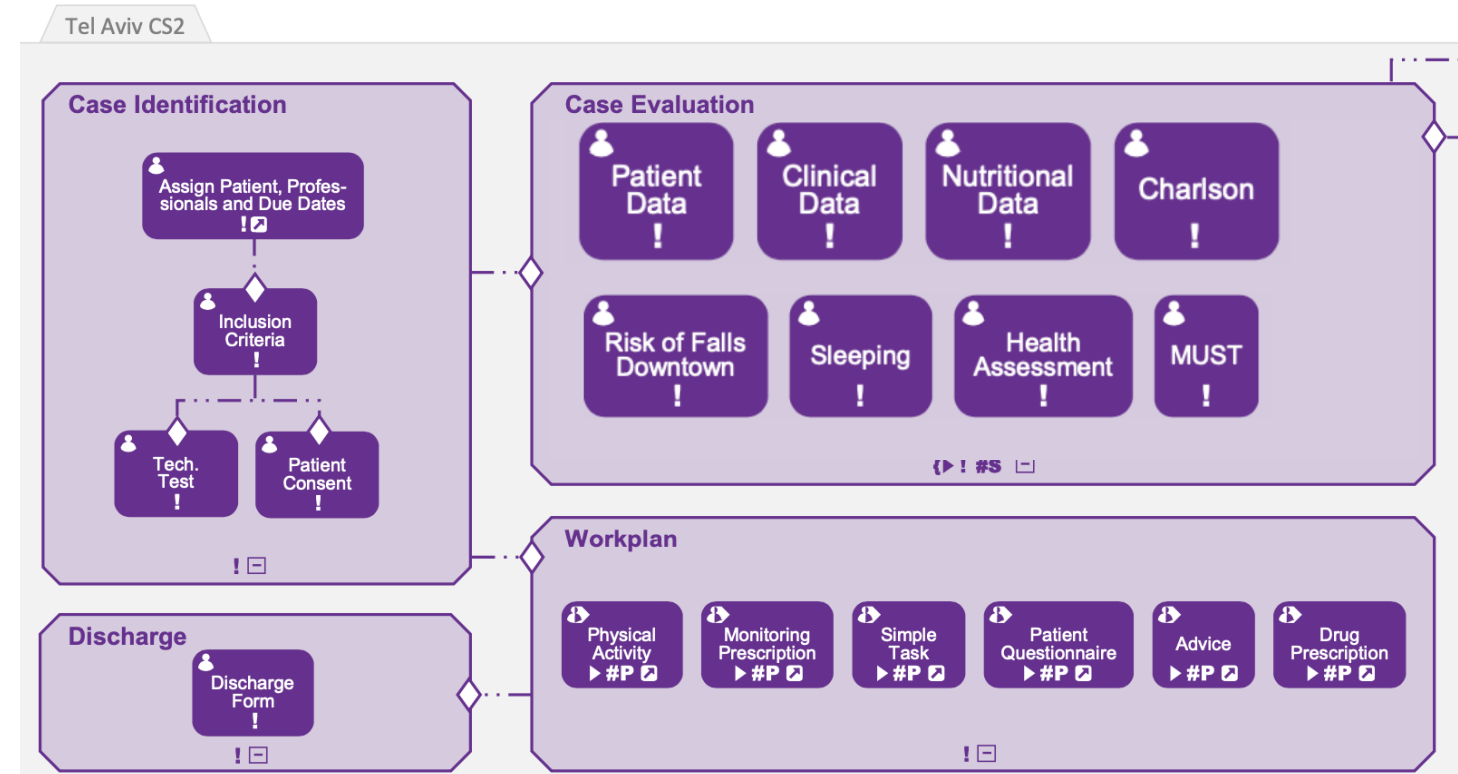
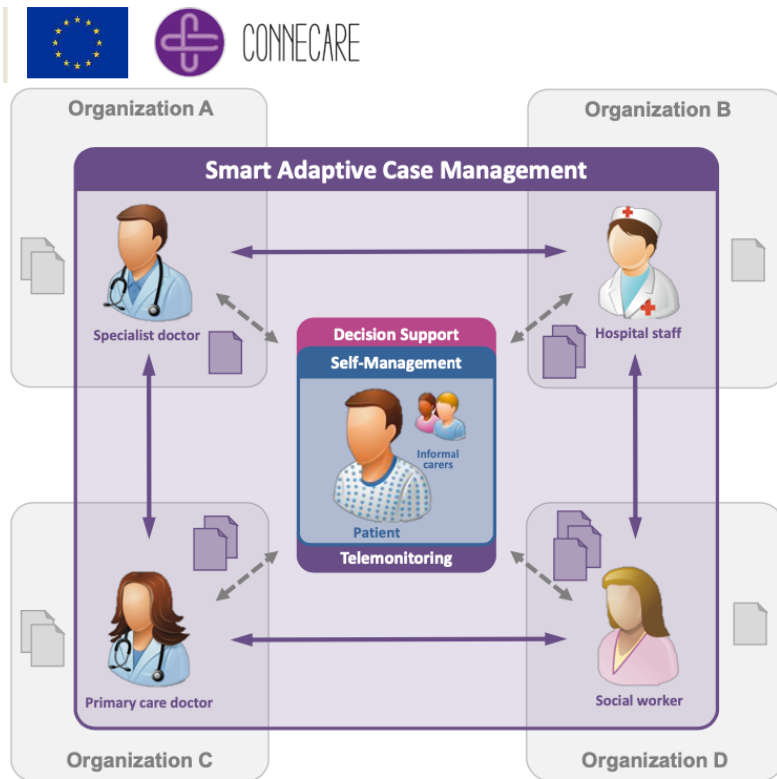
Evaluation



Timeline

Motivation

- Personalised Connected Care for Complex Chronic Patients (CONNECARE)
- Adaptive integrated care system for chronic care management
- Collaborative, meta-model-based, customizable solutions



Sample COPD treatment plan[1]

Current XML based modeling for a Stage

```
1 <StageDefinition
2   id="MCS2_CaseIdentification"
3   description="Case Identification"
4   isMandatory="true"
5   repeatable="ONCE"
6   entityDefinitionId="MCS2_Identification"
7   entityAttachPath="MCS2_Identification">
8
9   <!-- place <HumanTaskDefinition> here--->
10  <!-- place <AutomatedTaskDefinition> here--->
11  <!-- place <DualTaskDefinition> here--->
12
13 </StageDefinition>
14
15 <EntityDefinition id="MCS2_Identification" description="Identification">
16   <AttributeDefinition id="MCS2_Charlson" description="Charlson Comorbidity Index"
17     EntityDefinition.MCS2_Charlson" multiplicity="any" />
18   <AttributeDefinition id="MCS2_Barthel" description="Barthel" type="Link.EntityDef
19     multiplicity="any" />
20   <!-- Place additional <AttributeDefinition> to extend evaluation schema here -->
21 </EntityDefinition>
```



Acadela

(powered by TextX)

```
1 Stage Identification
2   #mandatory
3   #norepeat
4   owner = 'Setting.CaseOwner'
5   label = 'Identification'
```

Current XML based modeling for a Stage

```
1 <StageDefinition
2   id="MCS2_CaseIdentification"
3   description="Case Identification"
4   isMandatory="true"
5   repeatable="ONCE"
6   entityDefinitionId="MCS2_Identification"
7   entityAttachPath="MCS2_Identification">
8
9   <!-- place <HumanTaskDefinition> here--->
10  <!-- place <AutomatedTaskDefinition> here--->
11  <!-- place <DualTaskDefinition> here--->
12
13 </StageDefinition>

1 <EntityDefinition id="MCS2_Identification" description="Identification">
2   <AttributeDefinition id="MCS2_Charlson" description="Charlson Comorbidity Index"
3     EntityDefinition.MCS2_Charlson" multiplicity="any" />
4   <AttributeDefinition id="MCS2_Barthel" description="Barthel" type="Link.EntityDef
5     multiplicity="any" />
6   <!-- Place additional <AttributeDefinition> to extend evaluation schema here -->
7 </EntityDefinition>
```



Acadela

(powered by TextX)

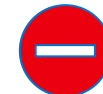
```
1 Stage Identification
2   #mandatory
3   #norepeat
4   owner = 'Setting.CaseOwner'
5   label = 'Identification'
```



Usability: more readable + user-friendly



Learnability: Intuitive grammar



Error handling in TextX is not user friendly

Syntax Errors

- TextX has built-in syntax error handler
- Technical terminology is non-intuitive, technical-oriented
- Some cannot be catch and some cannot be understood
- Doesn't tell cause of the error

Semantic Errors

- Cannot detect logical errors
- Cannot validate unexpected input (wrong referencing, type mismatch)

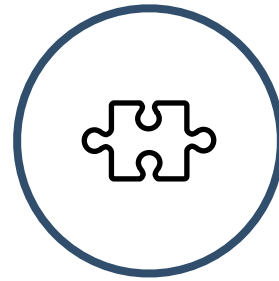
Goal: User-friendly and accurate error handler
Reducing learning and debugging effort



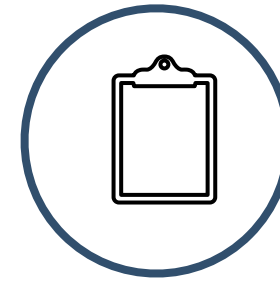
Motivation



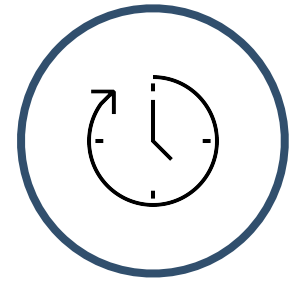
Research
Questions



Solution



Evaluation



Timeline

RQ1

What **syntactic** or **semantic errors** can be validated by the DSL during the modeling of clinical pathways?

RQ2

How to **identify and interpret the errors** to modelers in an **accurate, user-friendly manner**?

RQ3

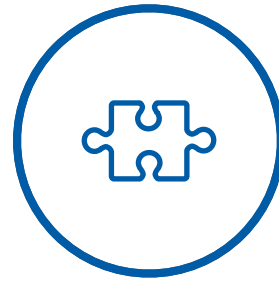
What is the **opinion of modelers** regarding the **usability and accuracy** of the error validator?



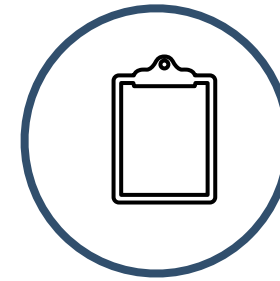
Motivation



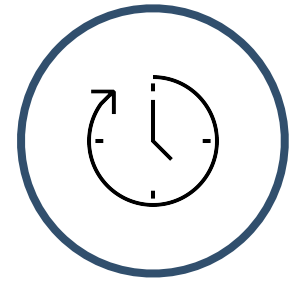
Research
Questions



Solution



Evaluation



Timeline

Syntax Errors

- Typos
 - Unexpected elements
 - Validating specific string patterns
- > Explaining error cause and suggestions

Syntax Errors

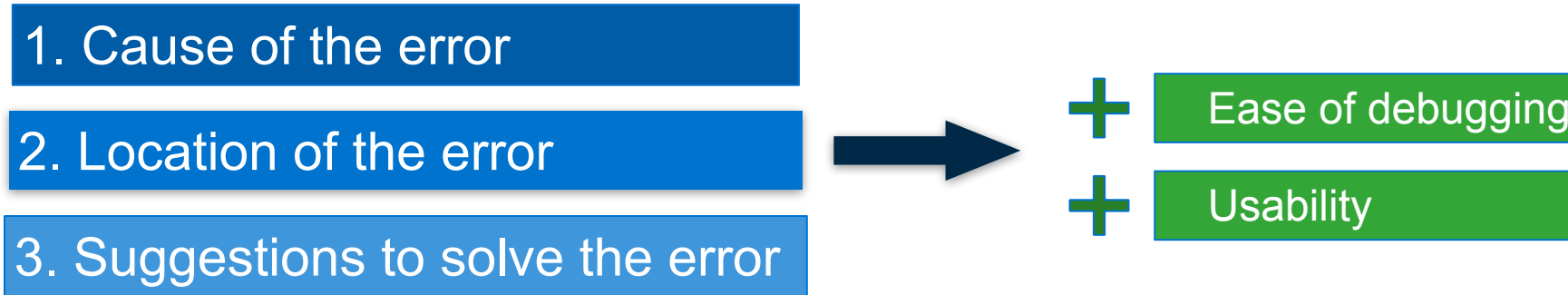
- Typos
 - Unexpected elements
 - Validating specific string patterns
- > Explaining error cause and suggestions

Semantic Errors

- Valid References / Paths
- Check precondition for valid Stage/Task Elements
- URL validation
- Unique ID for Stage, Task, Setting attributes and Fields

Developing a constraint validator to detect syntactic and semantic errors in the DSL Acadela

- More user-friendly and self-explanatory error messages
- Customizing and extending the syntax error handler of TextX
- Building a semantic error handling mechanism
- The algorithm also needs to consider edge cases where TextX fails to detect errors

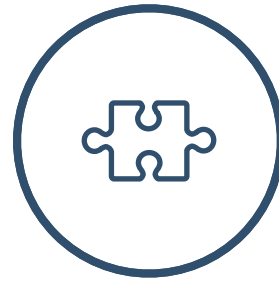




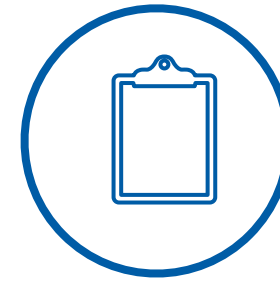
Motivation



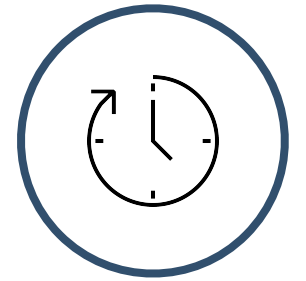
Research
Questions



Solution



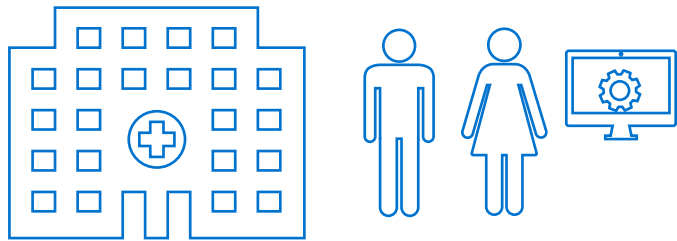
Evaluation



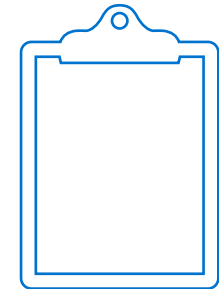
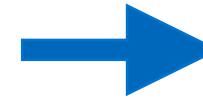
Timeline

Evaluation

- To evaluate the usability and accuracy of the validator



+



Test sessions with at least 5 modelers in medical facilities

Test sessions with at least 3 technical experts outside of medical field (extra)

Collecting quantitative and qualitative feedback

Evaluating the efficiency of new error messages with modelers in comparison to the original error messages

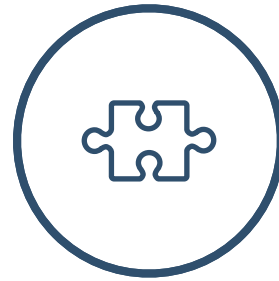
- Do they **understand** the error message?
- Were they able **to locate the error**?
- Were they able **to fix the error** easily using the error message?
- Overall user experience



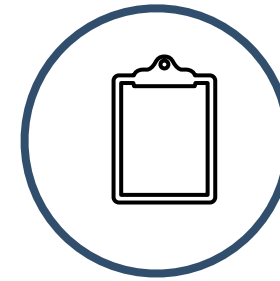
Motivation



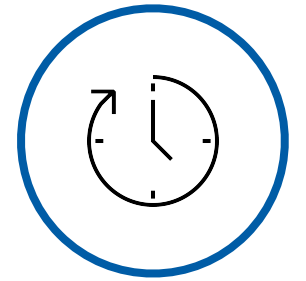
Research
Questions



Solution

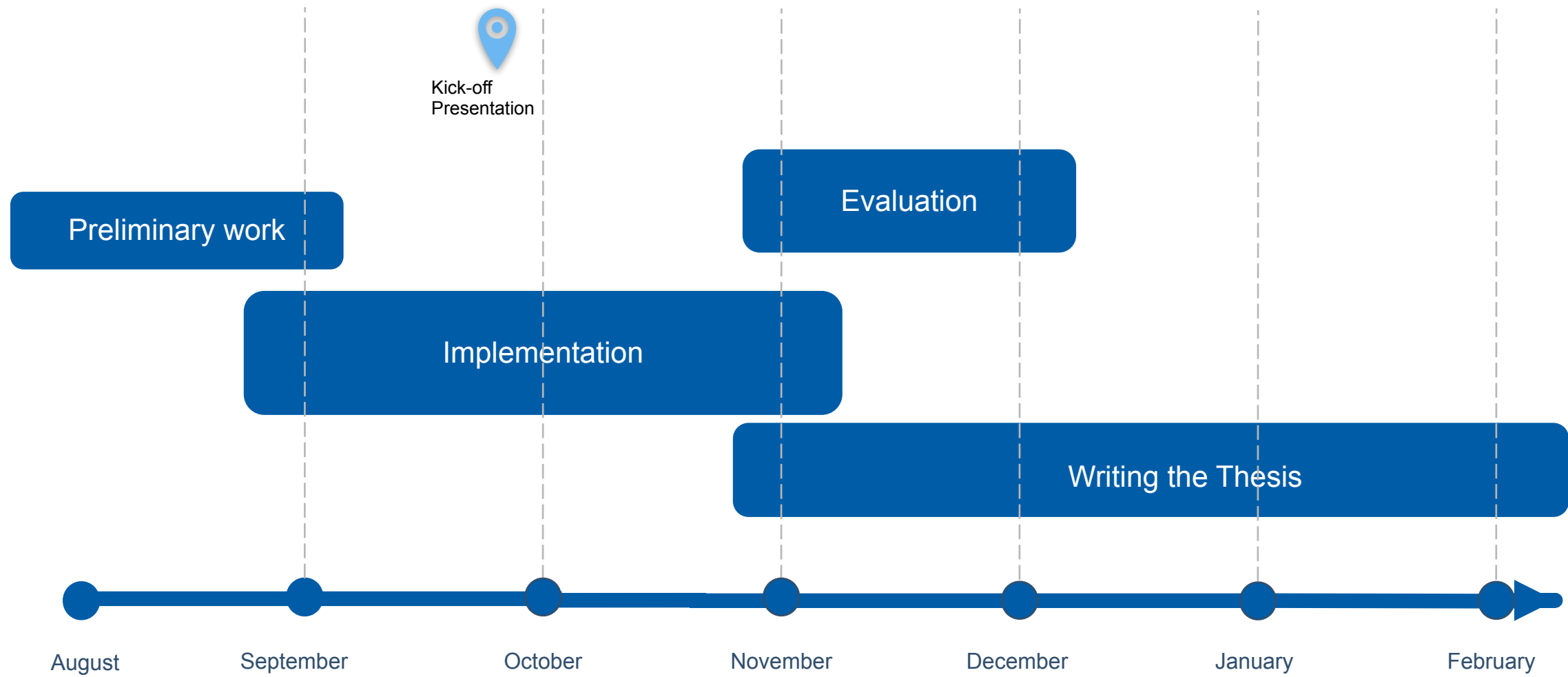


Evaluation



Timeline

Timeline





Selin Erdem

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business
Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.
Fax +49.89.289.17136 17132

www.matthes.in.tum.de
matthes@in.tum.de

