A formal Approach for Interfaces and Requirements of Smart Objects in Building Models

CIB W78 conference 2021, Luxembourg

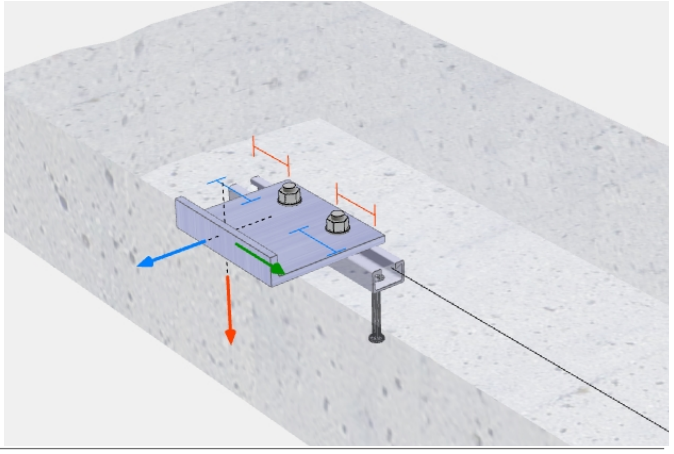
Dr.-Ing. Jakob Kirchner Prof. Dr.-Ing. Wolfgang Huhnt

Fachgebiet Bauinformatik
Institut für Bauingenieurwesen
Technische Universität Berlin, Germany

October 14th 2021

Motivation

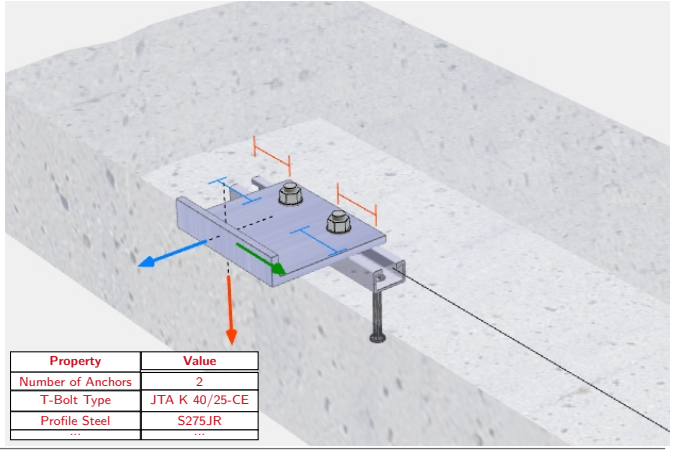
Smart object



Motivation

Smart object

- Unit of data

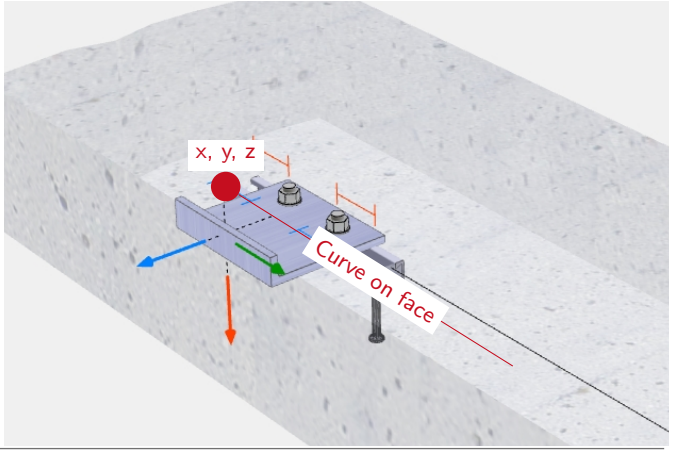


Property	Value
Number of Anchors	2
T-Bolt Type	JTA K 40/25-CE
Profile Steel	S275JR
...	...

Motivation

Smart object

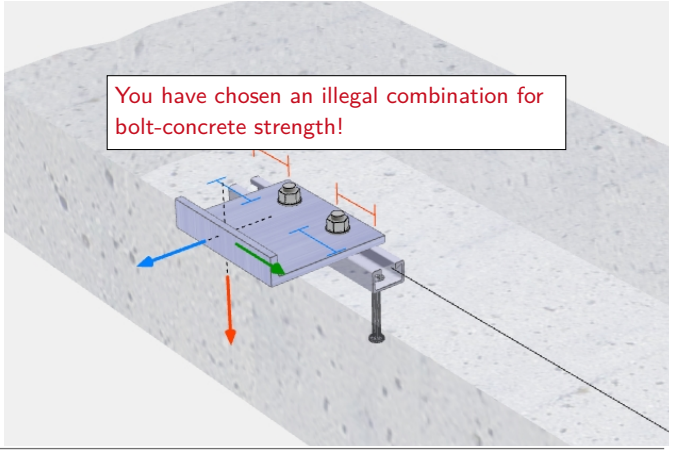
- Unit of data
- Relations



Motivation

Smart object

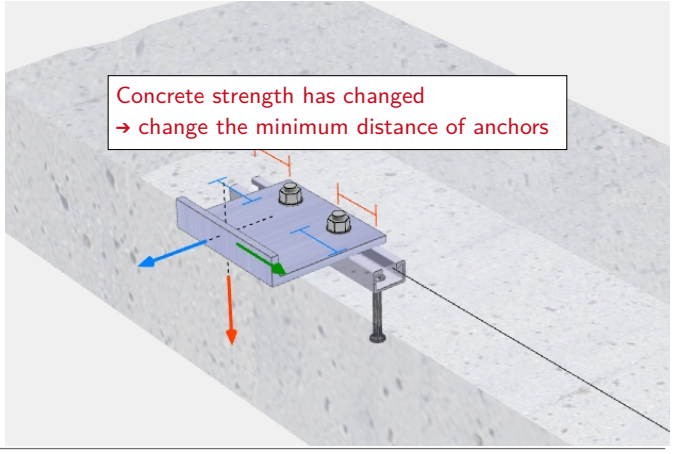
- Unit of data
- Relations
- Feedback



Motivation

Smart object

- Unit of data
- Relations
- Feedback
- Adaptive



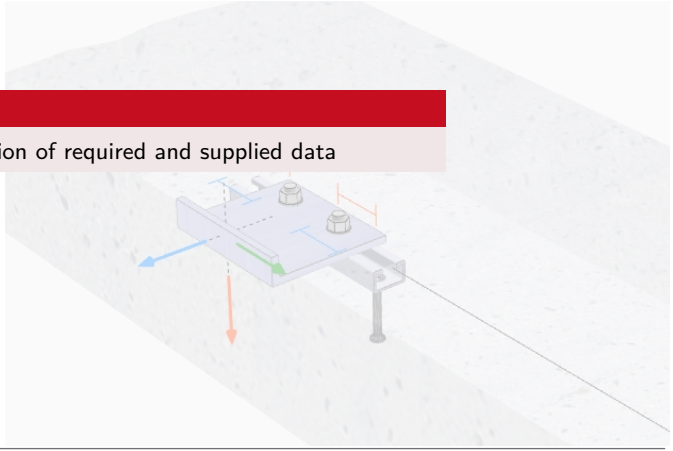
Motivation

Smart object

- Unit of data
- Relations
- Feedback
- Adaptive

Current problem

No generalized description of required and supplied data



Current software

Interaction points:

Current software

Interaction points:

- Simple geometry

Current software

Interaction points:

- Simple geometry
- Predefined variables

Current software

Interaction points:

- Simple geometry
- Predefined variables

Current software

Interaction points:

- Simple geometry
- Predefined variables

Problems:

Current software

Interaction points:

- Simple geometry
- Predefined variables

Problems:

- Barely expressive

Current software

Interaction points:

- Simple geometry
- Predefined variables

Problems:

- Barely expressive
- No customization

Current software

Interaction points:

- Simple geometry
- Predefined variables

Problems:

- Barely expressive
- No customization

Automated Model checking

Required Data:

Automated Model checking

Required Data:

Class 1: explicitly available

Automated Model checking

Required Data:

Class 1: explicitly available ✔

Automated Model checking

Required Data:

Class 1: explicitly available ✔

Class 2: derived values

Automated Model checking

Required Data:

Class 1: explicitly available ✔

Class 2: derived values ✔

Automated Model checking

Required Data:

Class 1: explicitly available ✔

Class 2: derived values ✔

Class 3: extended datastructures

Automated Model checking

Required Data:

- Class 1: explicitly available ✔
- Class 2: derived values ✔
- Class 3: extended datastructures
- Class 4: adaptive

Automated Model checking

Required Data:

Class 1: explicitly available ✔

Class 2: derived values ✔

Class 3: extended datastructures

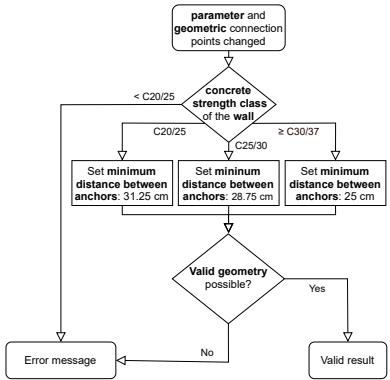
Class 4: adaptive (✔)

Automated Model checking

Required Data:

- Class 1: explicitly available ✔
- Class 2: derived values ✔
- Class 3: extended datastructures
- Class 4: adaptive (✔)

Smart objects for model checking:



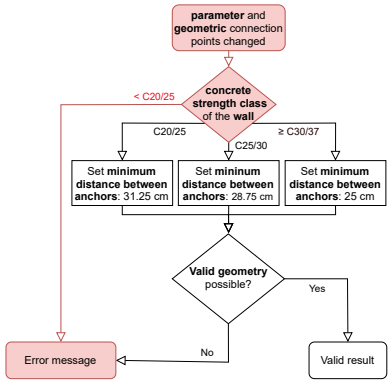
Automated Model checking

Required Data:

- Class 1: explicitly available ✔
- Class 2: derived values ✔
- Class 3: extended datastructures
- Class 4: adaptive (✔)

Smart objects for model checking:

- Feedback strategy



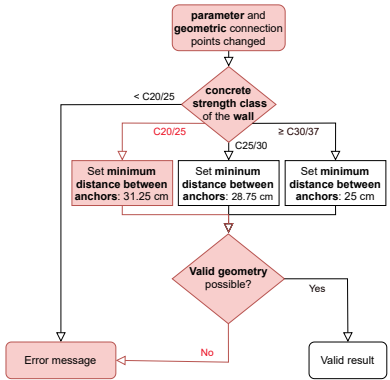
Automated Model checking

Required Data:

- Class 1: explicitly available ✔
- Class 2: derived values ✔
- Class 3: extended datastructures
- Class 4: adaptive (✔)

Smart objects for model checking:

- Feedback strategy



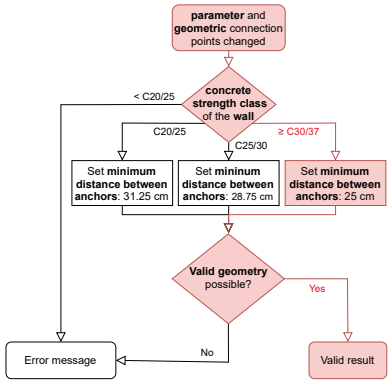
Automated Model checking

Required Data:

- Class 1: explicitly available ✔
- Class 2: derived values ✔
- Class 3: extended datastructures
- Class 4: adaptive (✔)

Smart objects for model checking:

- Feedback strategy
- Adaption strategy



Concept

- Availability of all data needed for a checking domain

Concept

- Availability of all data needed for a checking domain
- Seperation of each checking domain: *Ports*: 

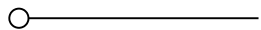
Concept

- Availability of all data needed for a checking domain
- Seperation of each checking domain: *Ports*: 
- providing and ... 

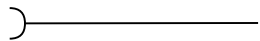
Concept

- Availability of all data needed for a checking domain
- Seperation of each checking domain: *Ports*:

- providing and ...



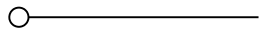
- ...requiring interface:



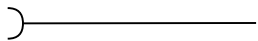
Concept

- Availability of all data needed for a checking domain
- Seperation of each checking domain: *Ports*:

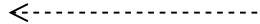
- providing and ...



- ...requiring interface:



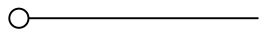
- Property-Access / Geometry-Access:



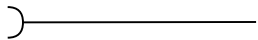
Concept

- Availability of all data needed for a checking domain
- Seperation of each checking domain: *Ports*:

- providing and ...



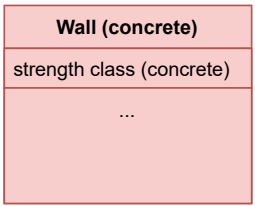
- ...requiring interface:

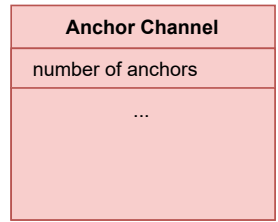


- Property-Access / Geometry-Access:

- "'Plug-In"'-System

Example

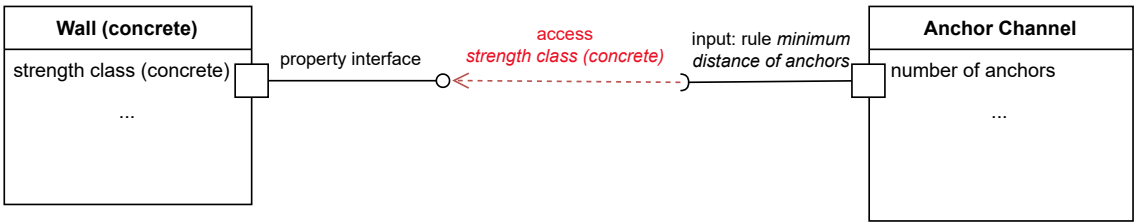

Wall (concrete)
strength class (concrete)
...


Anchor Channel
number of anchors
...

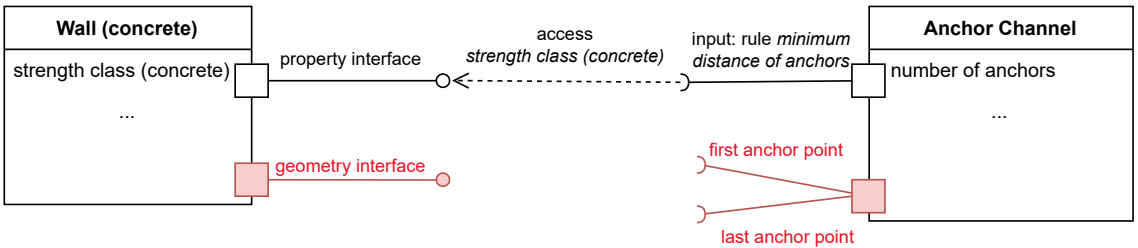
Example


Wall (concrete)
strength class (concrete)
...
property interface
input: rule *minimum distance of anchors*
Anchor Channel
number of anchors
...

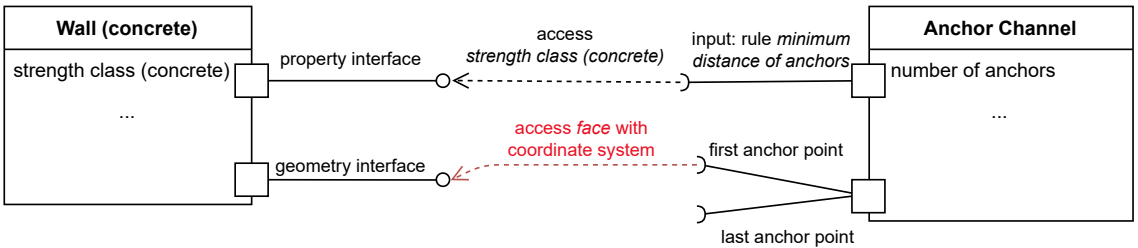
Example


Wall (concrete)
strength class (concrete)
...
property interface
access
strength class (concrete)
input: rule minimum
distance of anchors
Anchor Channel
number of anchors
...

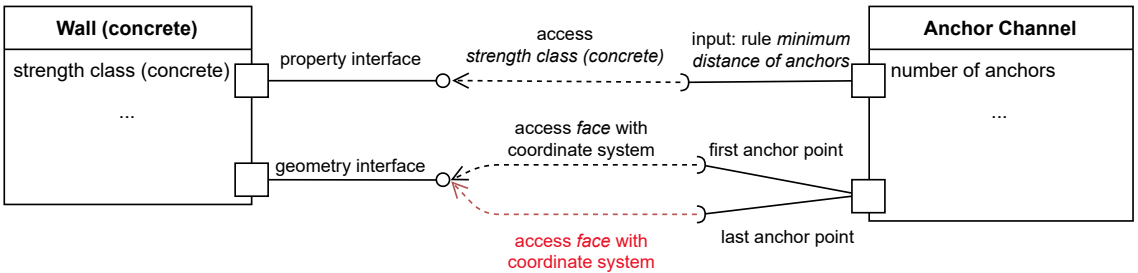
Example


Wall (concrete)
strength class (concrete)
...
property interface
geometry interface
access
strength class (concrete)
input: rule minimum distance of anchors
Anchor Channel
number of anchors
...
first anchor point
last anchor point

Example


Wall (concrete)
strength class (concrete)
...
property interface
geometry interface
access
strength class (concrete)
access face with
coordinate system
input: rule minimum
distance of anchors
Anchor Channel
number of anchors
...
first anchor point
last anchor point

Example


Wall (concrete)
strength class (concrete)
...
property interface
geometry interface
access
strength class (concrete)
access face with
coordinate system
access face with
coordinate system
input: rule minimum
distance of anchors
Anchor Channel
number of anchors
...
first anchor point
last anchor point

Conclusion

Conclusion

Accomplishments:

Conclusion

Accomplishments:

- Seperation of each checking domain

Conclusion

Accomplishments:

- Seperation of each checking domain
- Clear definition: Provided / Required

Conclusion

Accomplishments:

- Seperation of each checking domain
- Clear definition: Provided / Required
- General usage for elements / objects

Conclusion

Accomplishments:

- Seperation of each checking domain
- Clear definition: Provided / Required
- General usage for elements / objects

Conclusion

Accomplishments:

- Seperation of each checking domain
- Clear definition: Provided / Required
- General usage for elements / objects

Challanges:

Conclusion

Accomplishments:

- Seperation of each checking domain
- Clear definition: Provided / Required
- General usage for elements / objects

Challanges:

- Identification of properties

Conclusion

Accomplishments:

- Seperation of each checking domain
- Clear definition: Provided / Required
- General usage for elements / objects

Challanges:

- Identification of properties
- Identifying geometry objects

Outlook

Ideas:

Outlook

Ideas:

- Combining the approach with bSDD: Identification of properties

Outlook

Ideas:

- Combining the approach with bSDD: Identification of properties
- Constraining the requirements

Outlook

Ideas:

- Combining the approach with bSDD: Identification of properties
- Constraining the requirements
- Model/Software interface for Class 3 rules: Complex queries and checking

Outlook

Ideas:

- Combining the approach with bSDD: Identification of properties
- Constraining the requirements
- Model/Software interface for Class 3 rules: Complex queries and checking

Outlook

Ideas:

- Combining the approach with bSDD: Identification of properties
- Constraining the requirements
- Model/Software interface for Class 3 rules: Complex queries and checking

Next steps:

Outlook

Ideas:

- Combining the approach with bSDD: Identification of properties
- Constraining the requirements
- Model/Software interface for Class 3 rules: Complex queries and checking

Next steps:

- Reference implementation

Outlook

Ideas:

- Combining the approach with bSDD: Identification of properties
- Constraining the requirements
- Model/Software interface for Class 3 rules: Complex queries and checking

Next steps:

- Reference implementation
- Persistent and consistent geometry object classification

Outlook

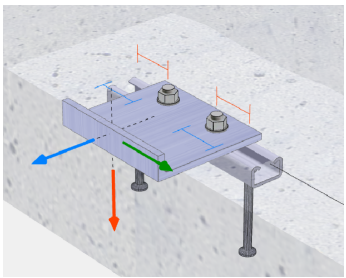
Ideas:

- Combining the approach with bSDD: Identification of properties
- Constraining the requirements
- Model/Software interface for Class 3 rules: Complex queries and checking

Next steps:

- Reference implementation
- Persistent and consistent geometry object classification

"'A formal Approach for Interfaces and Requirements of Smart Objects in Building Models"'



CIB W78 conference 2021, Luxembourg

Authors: **Dr.-Ing. Jakob Kirchner** Prof. Dr.-Ing. Wolfgang Huhnt

Feel free to contact
Mail: jakob.kirchner@tu-berlin.de
Phone: +49 30 314 72310
Chair: Bauinformatik
University: TU Berlin