Automated Consistency Analysis for Legal Contracts
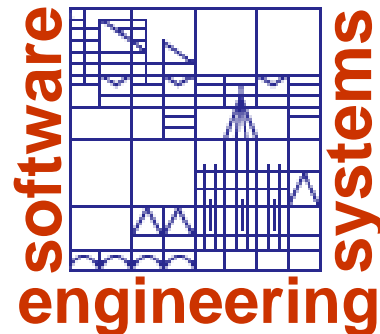
Alan Khoja[1], **Martin Kölbl**[2],
Stefan Leue[2], Rüdiger Wilhelmi[1]

[1]Chair for Civil Law
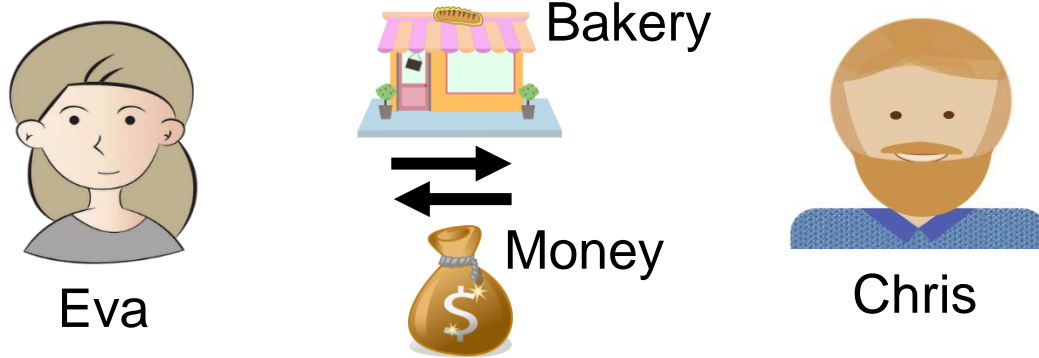[2]Chair for Software and Systems Engineering
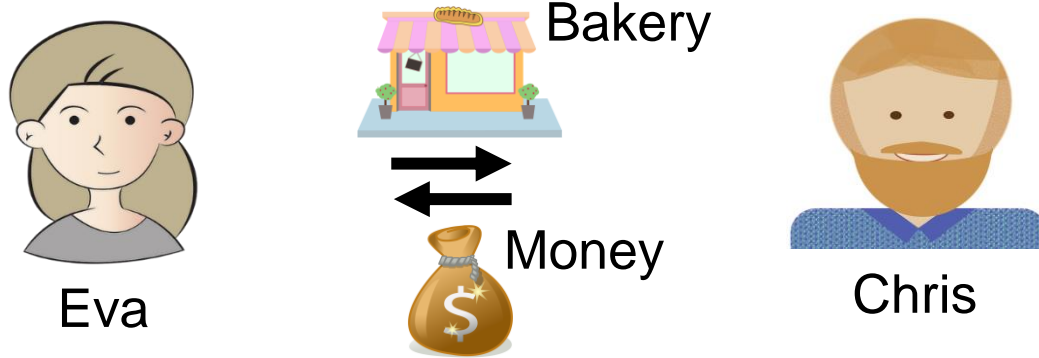
University of Konstanz

software engineering systems

Motivation

♦ **Sales & Purchase Agreement (SPA)**

Bakery

Eva

Money

Chris

software
engineering systems

Motivation

♦ **Sales & Purchase Agreement (SPA)**

Eva   Bakery   Money   Chris

§ 1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

§ 2 The Seller hereby represents and warrants to the Purchaser in the form of an independent guarantee pursuant to Section 311 (1) of the German Civil Code and exclusively in accordance with the provisions of this Agreement that the following statements (the ``Warranties") are true and correct as of the date of this Agreement and that the warranties set forth in this paragraph will also be true and correct as of the Closing Date:

2.1 The company can produce at least the 10.000 of Pretzels every day (PretzelWarranty).

§ 3 The Purchaser's rights arising from any inaccuracy of any of the Warranties contained in §1 shall be limited to supplementary performance claims and compensation claims against the Seller, subject to the provisions of

3.1 In case the PretzelWarranty is not met and then the creditor may demand subsequent performance within 28 days from the debtor.
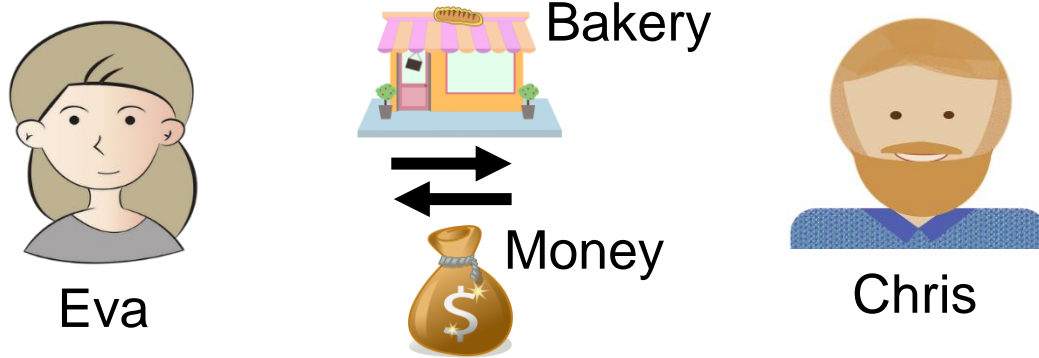
3.2 In case the PretzelWarranty is not met and the damage is above 1000 € then a compensation of 100€ per 100 pretzels not baked pretzels is paid within 14 days.

§ 4 Claims of §3 expire after 42 days.

§ 5 The Bakery AG is transferred by way of security to Bank

software
engineering
systems

Motivation

♦ **Sales & Purchase Agreement (SPA)**

Eva

Bakery

Money

Chris

§ 1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

§ 2 The Seller hereby represents and warrants to the Purchaser in the form of an independent guarantee pursuant to Section 311 (1) of the German Civil Code and exclusively in accordance with the provisions of this Agreement that the following statements (the ``Warranties") are true and correct as of the date of this Agreement and that the warranties set forth in this paragraph will also be true and correct as of the Closing Date:

2.1 The company can produce at least the 10.000 of Pretzels every day (PretzelWarranty).

§ 3 The Purchaser's rights arising from any inaccuracy of any of the Warranties contained in §1 shall be limited to supplementary performance claims and compensation claims against the Seller, subject to the provisions of

3.1 In case the PretzelWarranty is not met and then the cred[...] within 28 days from the debtor.

3.2 In case the PretzelWarranty is not met and the damage i[...]00€ per 100 pretzels not baked pretzels is paid within 14 days.
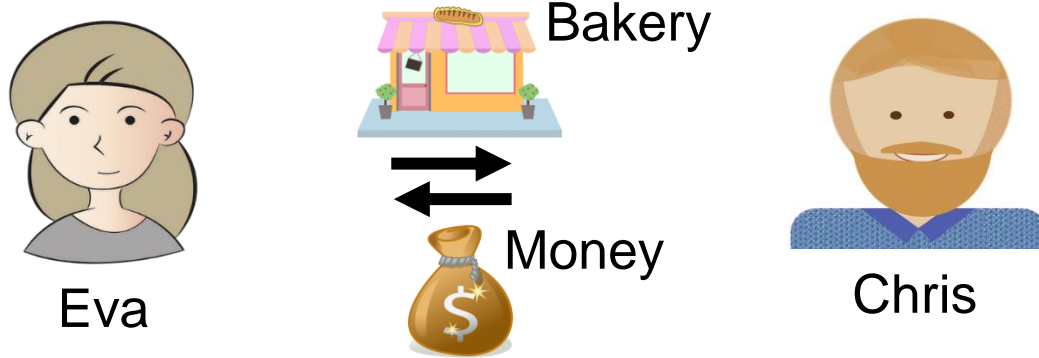
§ 4 Claims of §3 expire after 42 days.

§ 5 The Bakery AG is transferred by way of security to Bank[...]

> **§ claim**
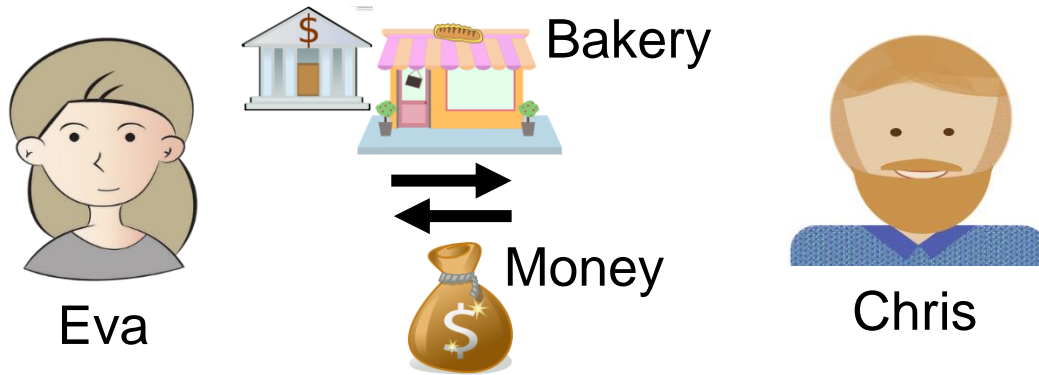> right to demand that another person does or refrains from an act

software engineering systems

Motivation

♦ **Sales & Purchase Agreement (SPA)**

Eva

Bakery

Money

Chris

§ Chris pays on date 28

§ Eva transfer bakery on date 28

Motivation

- **Sales & Purchase Agreement (SPA)**

Bakery

Eva   Money   Chris

§ Chris pays on date 28

§ Eva transfer bakery on date 28

– bakery owned by bank (security for debts)

Motivation

♦ **Sales & Purchase Agreement (SPA)**

Eva   Bakery   Money   Chris
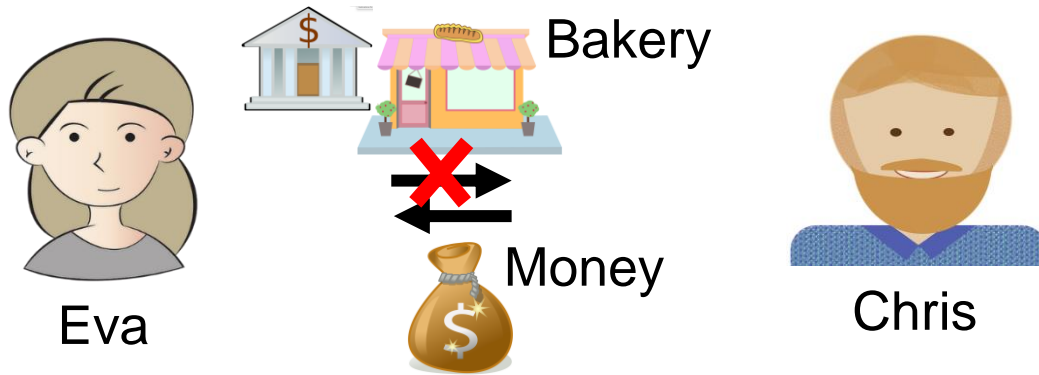
§ Chris pays on date 28

§ Eva transfer bakery on date 28

– bakery owned by bank (security for debts)

Motivation

♦ **Sales & Purchase Agreement (SPA)**

Bakery

Eva

Money

Chris

§ Chris pays on date 28

§ Eva transfer bakery on date 28

– bakery owned by bank (security for debts)

§ warranty: produce 10.000 pretzels

Motivation

♦ **Sales & Purchase Agreement (SPA)**

§ warranty: produce 10.000 pretzels

Motivation

- **Sales & Purchase Agreement (SPA)**
    - § warranty: produce 10.000 pretzels
        - – non-performance
            - § supplementary performance
            - § compensation
            - § withdraw

Motivation

♦ **Sales & Purchase Agreement (SPA)**

§ warranty: produce 10.000 pretzels

– non-performance

§ supplementary performance

§ compensation

§ withdraw

Motivation

♦ **Sales & Purchase Agreement (SPA)**
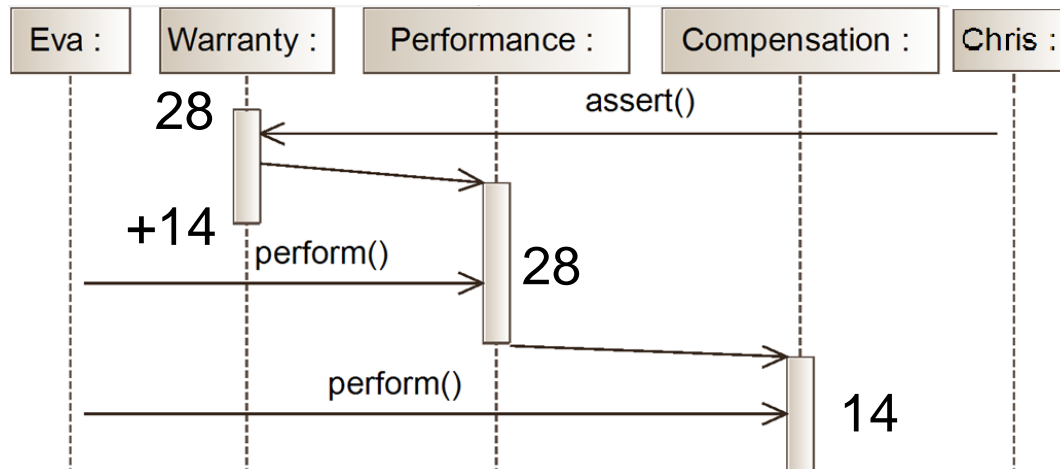
§ warranty: produce 10.000 pretzels

– non-performance

§ supplementary performance

§ compensation

§ withdraw

Motivation

- **Sales & Purchase Agreement (SPA)**
    - § warranty: produce 10.000 pretzels
        - – non-performance
            - § supplementary performance
            - § compensation
            - § withdraw

Motivation

- **Sales & Purchase Agreement (SPA)**
  - § warranty: produce 10.000 pretzels
    - – non-performance
      - § supplementary performance
      - § compensation
      - § withdraw

Motivation

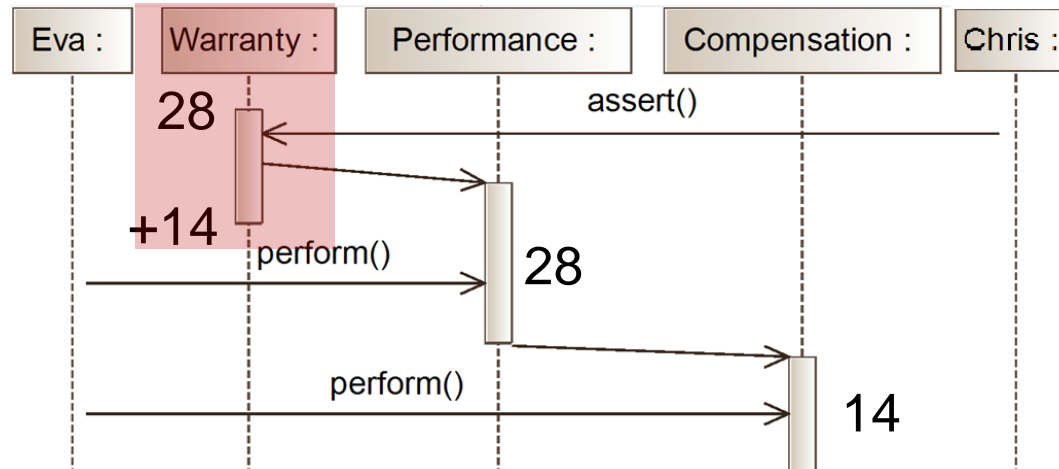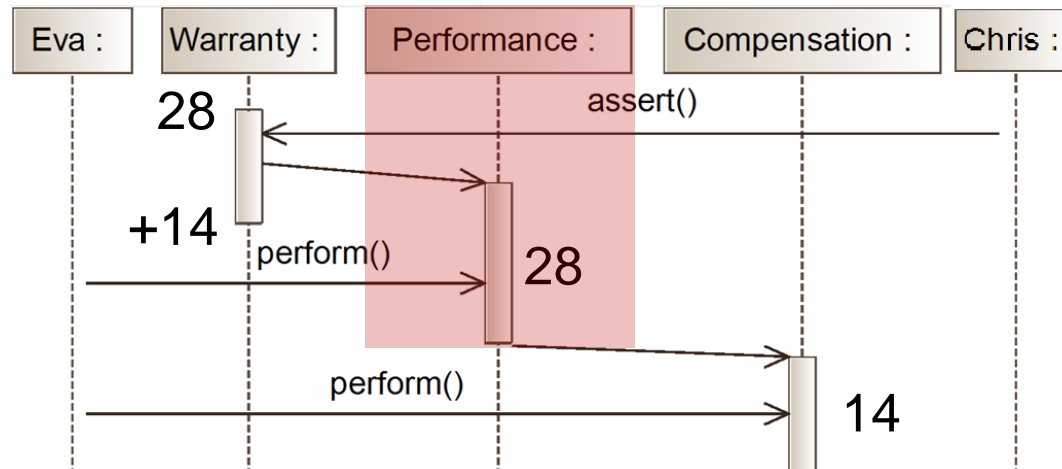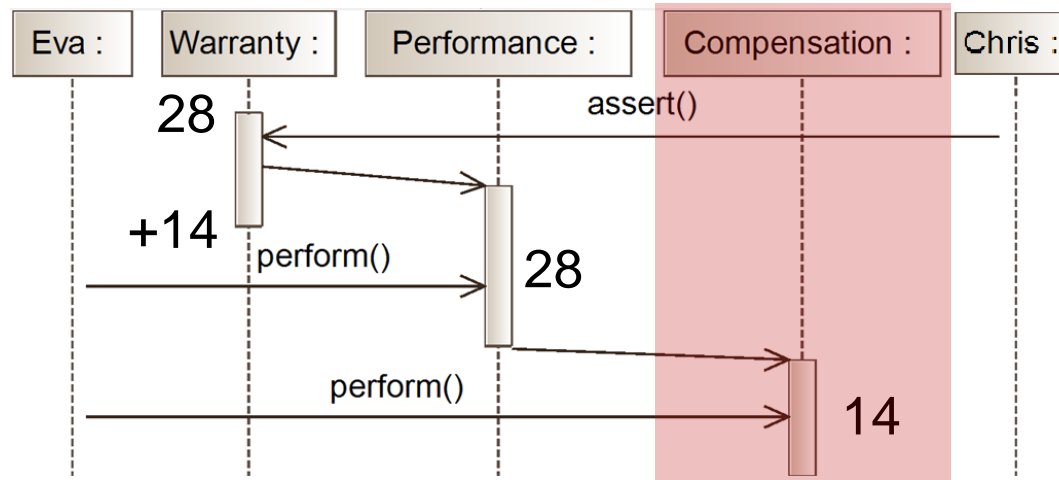♦ **Sales & Purchase Agreement (SPA)**

  § warranty: produce 10.000 pretzels

  – non-performance

    § supplementary performance

    § compensation

    § withdraw

Motivation

- **Sales & Purchase Agreement (SPA)**
  - § warranty: produce 10.000 pretzels
    - – non-performance
      - § supplementary performance
      - § compensation
      - § withdraw
    - – claim limitations

Motivation

♦ **Sales & Purchase Agreement (SPA)**

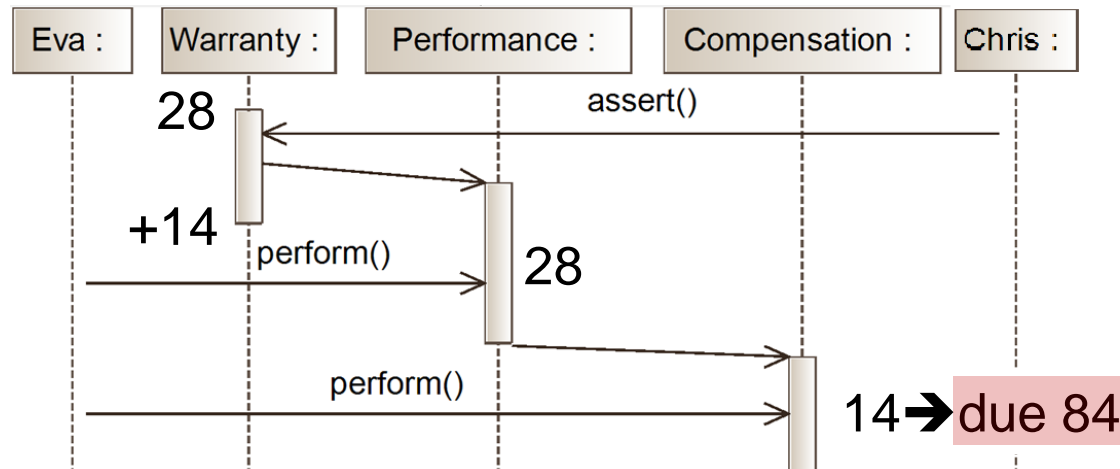§ warranty: produce 10.000 pretzels

– non-performance

§ supplementary performance

§ compensation

§ withdraw

– claim limitations

Automated Consistency Analyses

♦ **Why SPA?**

SPA

software engineering systems

Automated Consistency Analyses

♦ **Why SPA?**

▸ claims and consequences

▸ (mostly) independent of civil law

▸ 1xx pages with many details

SPA

software
engineering systems
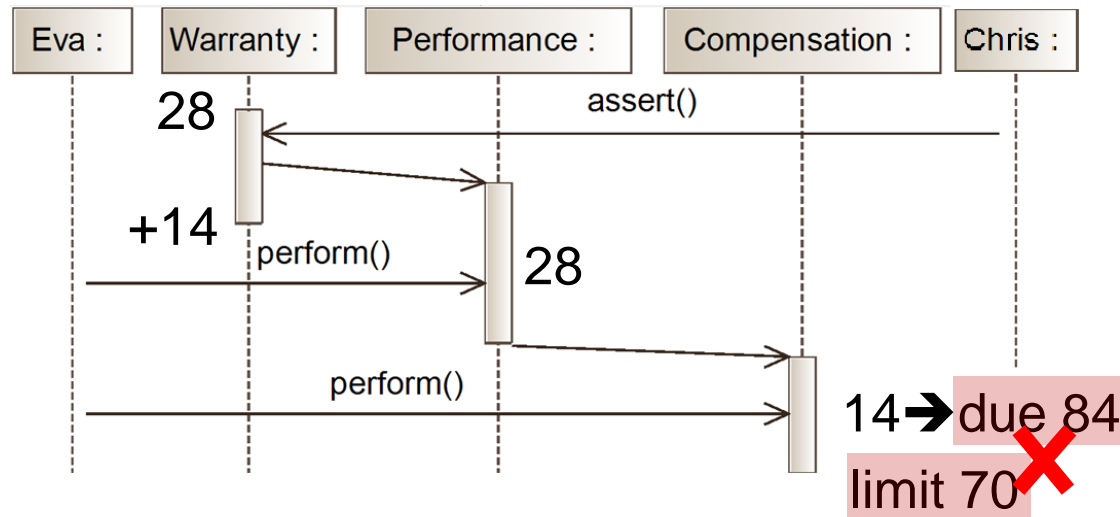
Automated Consistency Analyses

♦ **Why SPA?**

  ‣ claims and consequences

  ‣ (mostly) independent of civil law

  ‣ 1xx pages with many details

SPA

♦ **Check for (In)consistency**

  ‣ syntactic

  – important data contained?

  • buyer, seller, timing

  – superfluous elements?

  ‣ dynamic

  – claims can be performed

  – claim expires before due

software engineering systems

Automated Consistency Analyses

Automated Consistency Analyses

♦ **Tool ContractCheck**

▸

1. Blocks

SPA

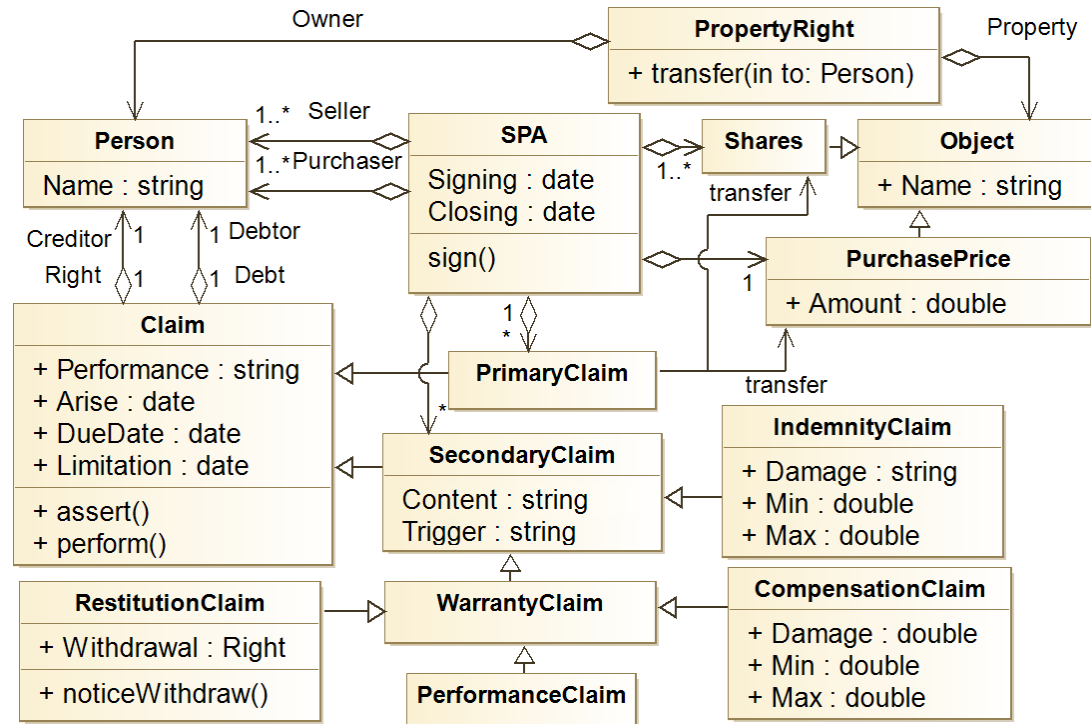|		
Owner	**PropertyRight**	Property
	+ transfer(in to: Person)	

Person
Name : string

SPA
Signing : date
Closing : date
sign()

Shares

Object
+ Name : string

1..* Seller
1..* Purchaser

PurchasePrice
+ Amount : double

Creditor 1
Right 1
1 Debtor
1 Debt

Claim
+ Performance : string
+ Arise : date
+ DueDate : date
+ Limitation : date
+ assert()
+ perform()

PrimaryClaim

SecondaryClaim
Content : string
Trigger : string

IndemnityClaim
+ Damage : string
+ Min : double
+ Max : double

RestitutionClaim
+ Withdrawal : Right
+ noticeWithdraw()

WarrantyClaim

CompensationClaim
+ Damage : double
+ Min : double
+ Max : double

PerformanceClaim

transfer
transfer

Automated Consistency Analyses

♦ **Tool ContractCheck**

1. Blocks — encode → 2. Object Diagram — formalize → 3. SMT Tool Lib Code — Z3 → 4. Results

SPA

Owner / PropertyRight / + transfer(in to: Person) / Property

Person / Name : string

1..* Seller / 1..* Purchaser

SPA / Signing : date / Closing : date / sign()

Shares / 1..* transfer

Object / + Name : string

PurchasePrice / + Amount : double

Creditor 1 / 1 Debtor / Right 1 / 1 Debt

Claim / + Performance : string / + Arise : date / + DueDate : date / + Limitation : date / + assert() / + perform()

PrimaryClaim / 1 * / transfer

SecondaryClaim / Content : string / Trigger : string

IndemnityClaim / + Damage : string / + Min : double / + Max : double

RestitutionClaim / + Withdrawal : Right / + noticeWithdraw()

WarrantyClaim

CompensationClaim / + Damage : double / + Min : double / + Max : double

PerformanceClaim

1. Parametrized Blocks

♦ **SPA**

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

…

1. Parametrized Blocks

♦ **SPA**

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

…

1. Parametrized Blocks

◆ SPA

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

…

◆ Parameterized Text Block

ID	
Text	
Objects	
Assign-ments	

1. Parametrized Blocks

♦ **SPA**

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.
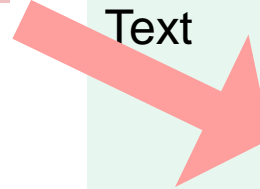
1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

…

♦ **Parameterized Text Block**

ID	Block1
Text	The Seller $seller.Name hereby sells the shares of $shares.Name with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser $purchaser.Name who accepts such sale.
Objects	
Assign-ments	

software engineering systems
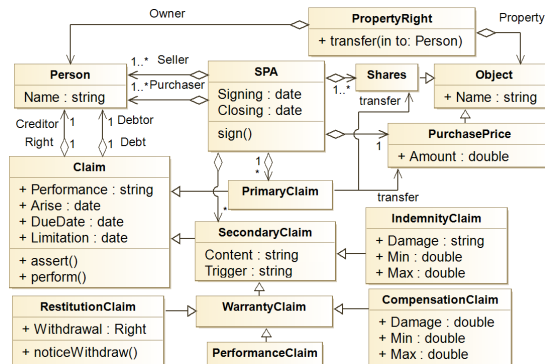
1. Parametrized Blocks

◆ SPA

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

…

◆ Parameterized Text Block

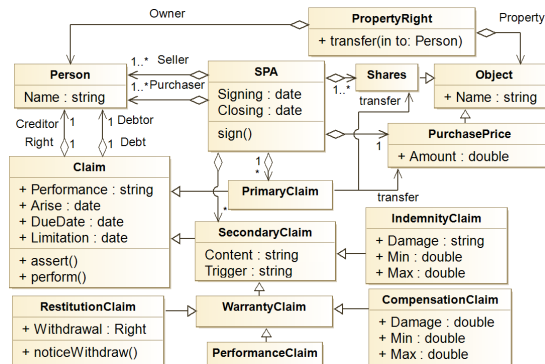| ID	Block1
Text	The Seller $seller.Name hereby sells the shares of $shares.Name with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser $purchaser.Name who accepts such sale.
Objects	„seller:Person"
Assign-ments	„seller.Name=Eva"

1. Parametrized Blocks

♦ SPA

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

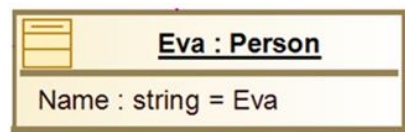1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

…

♦ Parameterized Text Block

ID	Block1
Text	The Seller $seller.Name hereby sells the shares of $shares.Name with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser $purchaser.Name who accepts such sale.
Objects	„seller:Person"
Assign-ments	„seller.Name=Eva"

software engineering systems
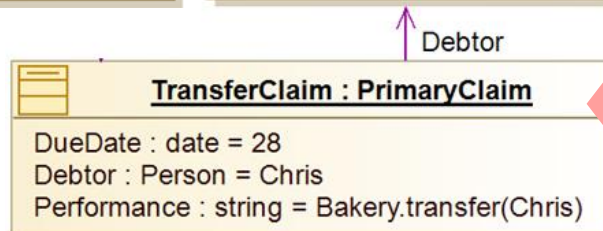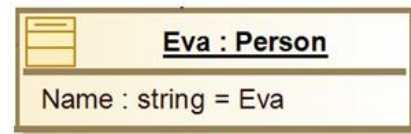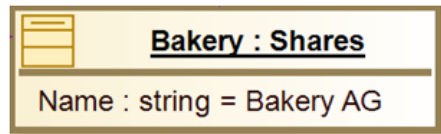
1. Parametrized Blocks

◆ SPA

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

…

◆ Parameterized Text Block

ID	Block1
Text	The Seller $seller.Name hereby sells the shares of $shares.Name with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser $purchaser.Name who accepts such sale.
Objects	„seller:Person", „shares:Share", „transfer:Claim"
Assign-ments	„seller.Name=Eva", „shares.Name=Bakery AG", „transfer=$shares.transfer()"

software engineering systems
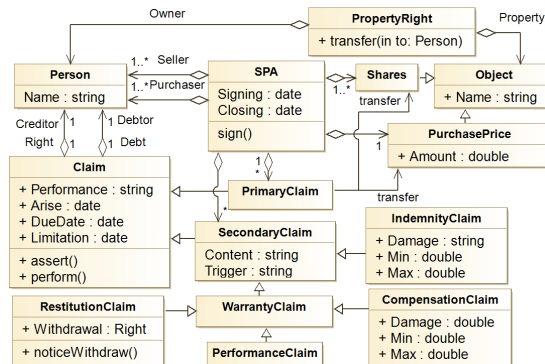
1. Parametrized Blocks

♦ **SPA**

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.

1.2 The purchaser pays the purchase price 40.000 € to the seller.

1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.

1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

Webinterface:

Block1 [Remove] [Hide] [Up] [Down]
The Seller [Eva] hereby sells shares of [Bakery AG] , with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser [Chris] , who accepts such sale.

[+]

Block2 [Remove] [Hide] [Up] [Down]
The purchaser pays the purchase price [40000]
€ to the seller on date [28] .

♦ **Parameterized Text Block**

ID	Block1
Text	The Seller $seller.Name hereby sells the shares of $shares.Name with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser $purchaser.Name who accepts such sale.
Objects	„seller:Person", „shares:Share", „transfer:Claim"
Assign-ments	„seller.Name=Eva", „shares.Name=Bakery AG", „transfer=$shares.transfer()"

software engineering systems

1. Parametrized Blocks

◆ SPA

§1 1.1 The Seller Eva hereby sells the shares of Bakery AG with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser Chris who accepts such sale.
1.2 The purchaser pays the purchase price 40.000 € to the seller.
1.3 If the TransferClaim is not performed, the claim.Debtor has the right to withdraw.
1.4 If the PayClaim is not performed, the claim.Debtor has the right to withdraw.

Webinterface:

Block1 [Remove] [Hide] [Up] [Down]
The Seller [Eva] hereby sells shares of [Bakery AG], with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser [Chris], who accepts such sale.

[+]

Block2 [Remove] [Hide] [Up] [Down]
The purchaser pays the purchase price [40000] € to the seller on date [28].

◆ Parameterized Text Block

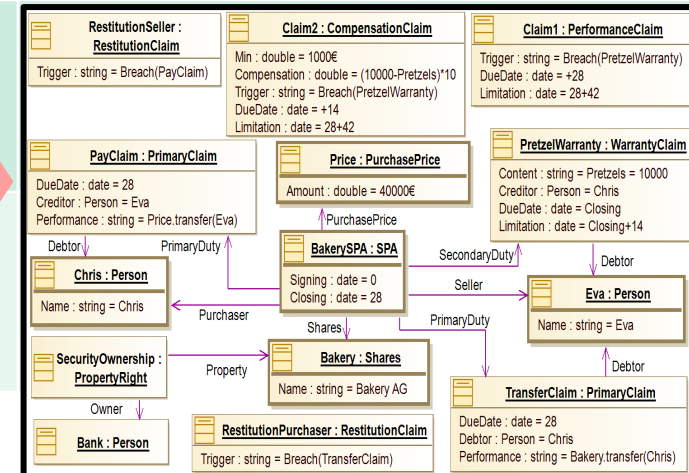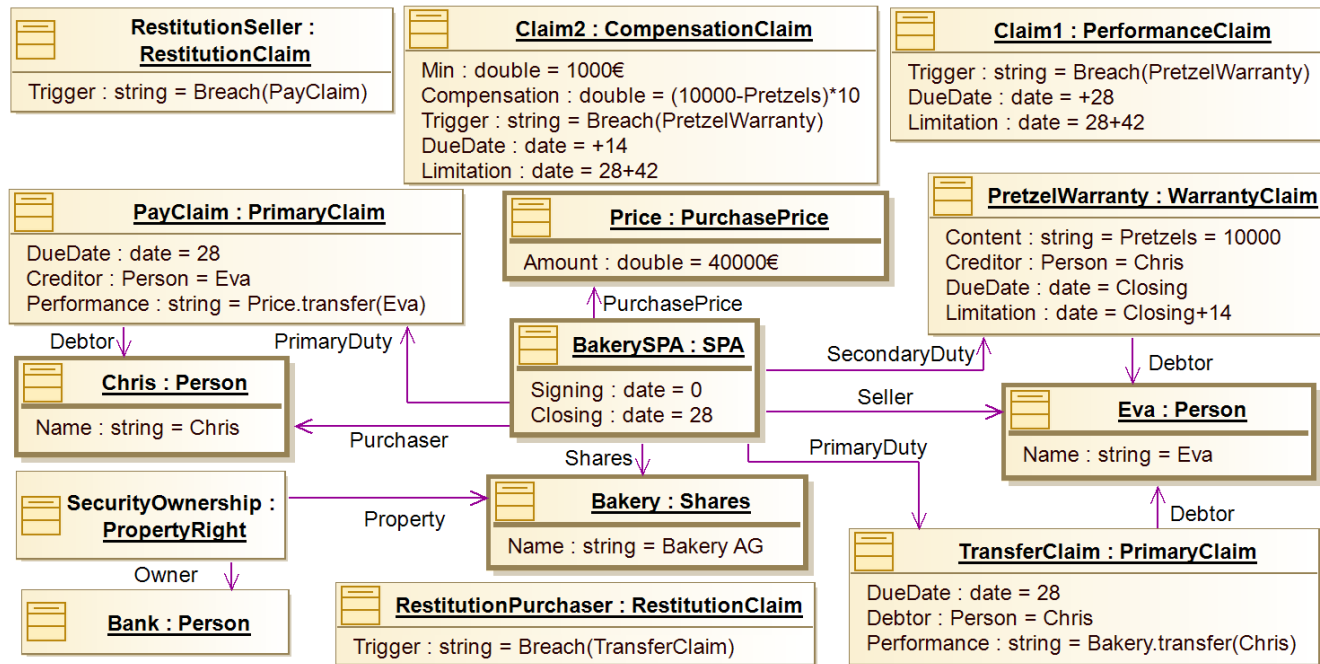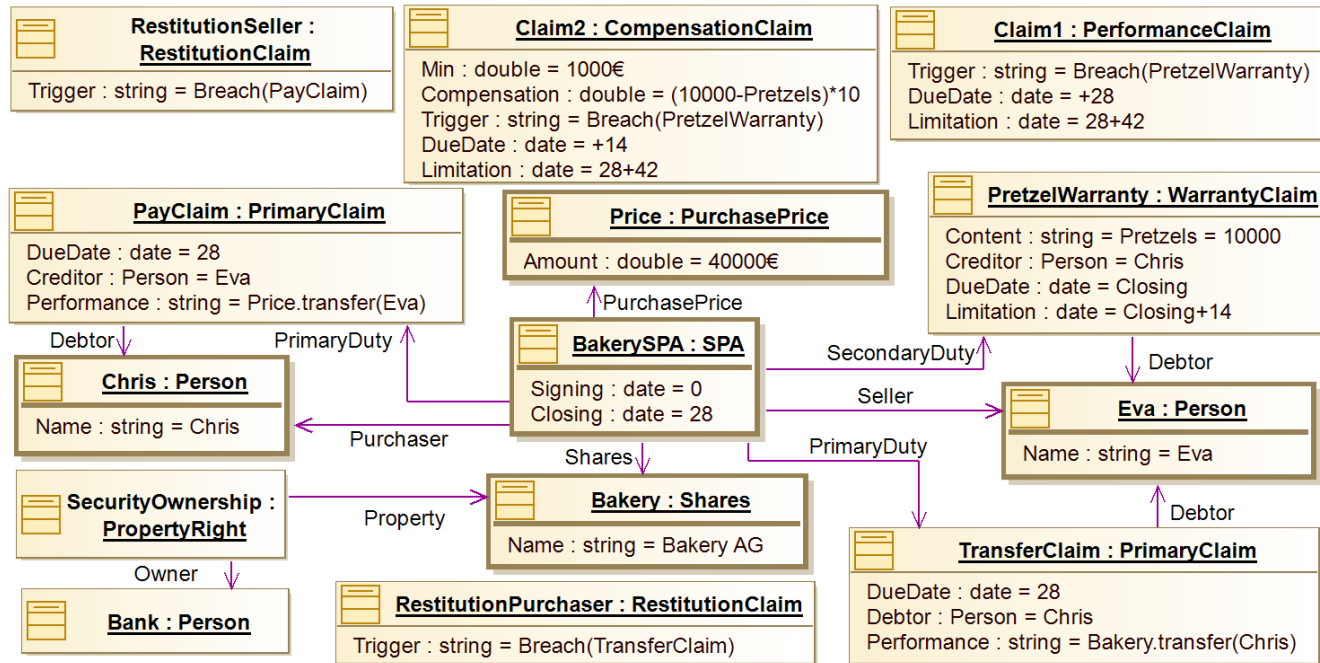| ID	Block1
Text	The Seller $seller.Name hereby sells the shares of $shares.Name with all rights and obligations pertaining thereto (including the dividend right for the current financial year), to the Purchaser $purchaser.Name who accepts such sale.
Objects	
Assig ments	

software engineering systems

2. Contract Object Diagram

♦ **Bakery SPA**

RestitutionSeller : RestitutionClaim
Trigger : string = Breach(PayClaim)

Claim2 : CompensationClaim
Min : double = 1000€
Compensation : double = (10000-Pretzels)*10
Trigger : string = Breach(PretzelWarranty)
DueDate : date = +14
Limitation : date = 28+42

Claim1 : PerformanceClaim
Trigger : string = Breach(PretzelWarranty)
DueDate : date = +28
Limitation : date = 28+42

PayClaim : PrimaryClaim
DueDate : date = 28
Creditor : Person = Eva
Performance : string = Price.transfer(Eva)

Price : PurchasePrice
Amount : double = 40000€

PretzelWarranty : WarrantyClaim
Content : string = Pretzels = 10000
Creditor : Person = Chris
DueDate : date = Closing
Limitation : date = Closing+14

BakerySPA : SPA
Signing : date = 0
Closing : date = 28

Chris : Person
Name : string = Chris

Eva : Person
Name : string = Eva

SecurityOwnership : PropertyRight

Bakery : Shares
Name : string = Bakery AG

TransferClaim : PrimaryClaim
DueDate : date = 28
Debtor : Person = Chris
Performance : string = Bakery.transfer(Chris)

Bank : Person

RestitutionPurchaser : RestitutionClaim
Trigger : string = Breach(TransferClaim)

♦ **Bakery SPA**

RestitutionSeller : RestitutionClaim
Trigger : string = Breach(PayClaim)

Claim2 : CompensationClaim
Min : double = 1000€
Compensation : double = (10000-Pretzels)*10
Trigger : string = Breach(PretzelWarranty)
DueDate : date = +14
Limitation : date = 28+42

Claim1 : PerformanceClaim
Trigger : string = Breach(PretzelWarranty)
DueDate : date = +28
Limitation : date = 28+42

PayClaim : PrimaryClaim
DueDate : date = 28
Creditor : Person = Eva
Performance : string = Price.transfer(Eva)

Price : PurchasePrice
Amount : double = 40000€

PretzelWarranty : WarrantyClaim
Content : string = Pretzels = 10000
Creditor : Person = Chris
DueDate : date = Closing
Limitation : date = Closing+14

Chris : Person
Name : string = Chris

BakerySPA : SPA
Signing : date = 0
Closing : date = 28

Eva : Person
Name : string = Eva

SecurityOwnership : PropertyRight

Bakery : Shares
Name : string = Bakery AG

TransferClaim : PrimaryClaim
DueDate : date = 28
Debtor : Person = Chris
Performance : string = Bakery.transfer(Chris)

Bank : Person

RestitutionPurchaser : RestitutionClaim
Trigger : string = Breach(TransferClaim)

Labels: Debtor, PrimaryDuty, PurchasePrice, SecondaryDuty, Seller, Debtor, Purchaser, Shares, PrimaryDuty, Property, Owner, Debtor

‣ syntactic analysis

– compare with class diagram

software engineering systems

♦ **Bakery SPA**

▸ syntactic analysis

– compare with class diagram

▸ dynamic analysis

– contradictions?, timing?

♦ **Logical Encoding in SMT**

♦ **Encoding in SMT**

  ‣ legal facts

   – person set $P := \{Eva=1, Chris=2, Bank=3\}$

   – object set $O := \{Bakery=1, Price=2\}$

   – dates d_s, d_c, d_T , …

♦ **Encoding in SMT**

  ‣ legal facts

    – person set $P := \{Eva=1, Chris=2, Bank=3\}$

    – object set $O := \{Bakery=1, Price=2\}$

    – dates d_s, d_c, d_T , …

    – owner function $\phi_{owner}^{B} \equiv owner(Bakery) = Bank$

3. Legal Encoding / Analysis

♦ **Encoding in SMT**

▸ legal facts

   – person set $P := \{Eva=1, Chris=2, Bank=3\}$

   – object set $O := \{Bakery=1, Price=2\}$

   – dates d_s, d_c, d_T, …

   – owner function $\phi^B_{owner} \equiv owner(Bakery) = Bank$

▸ claim encoding

   – legal claim $c \in C_I$ with date d_c with performance l_c

     • e.g. $\phi_T \equiv (d_T = -1) \lor (28 \leq d_T \leq 70 \Rightarrow owner(1) = 1)$

3. Legal Encoding / Analysis

♦ **Encoding in SMT**

  ‣ legal facts

   – person set $P \coloneqq$ {Eva=1, Chris=2, Bank=3}

   – object set $O \coloneqq$ {Bakery=1, Price=2}

   – dates d_s, d_c, d_T, …

   – owner function $\phi^B_{owner} \equiv owner(Bakery) = Bank$

  ‣ claim encoding

   – legal claim $c \in C_I$ with date d_c with performance l_c

    • e.g. $\phi_T \equiv (d_T = -1) \lor (28 \le d_T \le 70 \Rightarrow owner(1) = 1)$

software systems engineering

3. Legal Encoding / Analysis

♦ **Encoding in SMT**

▸ legal facts

– person set $P \coloneqq$ {Eva=1, Chris=2, Bank=3}

– object set $O \coloneqq$ {Bakery=1, Price=2}

– dates d_s, d_c, d_T, …

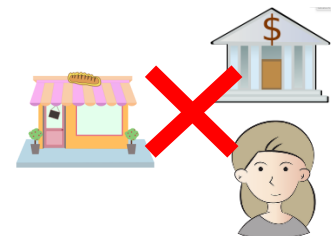– owner function $\phi^{B}_{owner} \equiv owner(Bakery) = Bank$

▸ claim encoding

– legal claim $c \in C_I$ with date d_c with performance l_c

• e.g. $\phi_T \equiv (d_T = -1) \lor (28 \leq d_T \leq 70 \Rightarrow owner(1) = 1)$

▸ claim satisfiability analysis

– $c \in C_I : \Phi_c \equiv \phi_{owner} \land \phi_c \land d_c \geq 0$

♦ **Encoding in SMT**

▸ legal facts

  – person set $P := \{Eva=1, Chris=2, Bank=3\}$

  – object set $O := \{Bakery=1, Price=2\}$

  – dates d_s, d_c, d_T, …

  – owner function $\phi_{owner}^{B} \equiv owner(Bakery) = Bank$
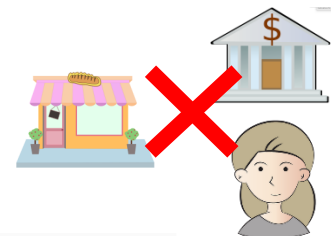
▸ claim encoding

  – legal claim $c \in C_I$ with date d_c with performance l_c

    • e.g. $\phi_T \equiv (d_T = -1) \vee (28 \leq d_T \leq 70 \Rightarrow owner(1) = 1)$

▸ claim satisfiability analysis

  – $c \in C_I : \Phi_c \equiv \phi_{owner} \wedge \phi_c \wedge d_c \geq 0$

  – e.g. $\Phi_T \equiv owner(1) = 3 \wedge ((d_T = -1) \vee 28 \leq d_T \Rightarrow owner(1) = 1) \wedge d_T \geq 0$

3. Legal Encoding / Analysis

♦ **Encoding in SMT**

▸ legal facts

 – person set $\mathrm{P} := \{\text{Eva}=1, \text{Chris}=2, \text{Bank}=3\}$

 – object set $\mathrm{O} := \{\text{Bakery}=1, \text{Price}=2\}$

 – dates $d_s, d_c, d_T, \ldots$

 – owner function $\quad \phi_{owner}^B \equiv owner(Bakery) = Bank$

▸ claim encoding

 – legal claim $c \in C_I$ with date d_c with performance l_c

  • e.g. $\phi_T \equiv (d_T = -1) \vee (28 \leq d_T \leq 70 \Rightarrow owner(1) = 1)$

▸ claim satisfiability analysis

 – $c \in C_I : \Phi_c \equiv \phi_{owner} \wedge \phi_c \wedge d_c \geq 0$

 – e.g. $\Phi_T \equiv owner(1) = 3 \wedge ((d_T = -1) \vee$
 $28 \leq d_T \Rightarrow owner(1) = 1) \wedge d_T \geq 0$

software engineering systems

3. Legal Encoding / Analysis

♦ **Encoding in SMT**

  ‣ legal facts

  – person set $P := \{Eva=1, Chris=2, Bank=3\}$

  – object set $O := \{Bakery=1, Price=2\}$

  – dates $d_s, d_c, d_T, \ldots$

  – owner function $\phi_{owner}^B \equiv owner(Bakery) = Bank$

  ‣ claim encoding

  – legal claim $c \in C_I$ with date d_c with performance l_c

  • e.g. $\phi_T \equiv (d_T = -1) \vee (28 \leq d_T \leq 70 \Rightarrow owner(1) = 1)$

  ‣ claim satisfiability analysis

  – $c \in C_I : \Phi_c \equiv \phi_{owner} \wedge \phi_c \wedge d_c \geq 0$

  – e.g. $\Phi_T \equiv owner(1) = 3 \wedge ((d_T = -1) \vee$
  $28 \leq d_T \Rightarrow owner(1) = 1) \wedge d_T \geq 0$

3. Legal Encoding / Analysis

♦ **Encoding in SMT**

▸ legal facts

 – person set $P := \{Eva=1, Chris=2, Bank=3\}$

 – object set $O := \{Bakery=1, Price=2\}$

 – dates d_s, d_c, d_T, …

 – owner function $\phi_{owner}^B \equiv owner(Bakery) = Bank$

▸ claim encoding

 – legal claim $c \in C_I$ with date d_c with performance l_c

  • e.g. $\phi_T \equiv (d_T = -1) \lor (28 \le d_T \le 70 \Rightarrow owner(1) = 1)$

▸ claim satisfiability analysis

 – $c \in C_I : \Phi_c \equiv \phi_{owner} \land \phi_c \land d_c \ge 0$

 – e.g. $\Phi_T \equiv owner(1) = 3 \land ((d_T = -1) \lor$
  $28 \le d_T \Rightarrow owner(1) = 1) \land d_T \ge 0$

▸ tool result

Block1:
The seller Eva sells the shares of the company Bakery AG to purchaser Chris for 40000 €.

Block9:
The Bakery AG is transferred by way of security to Bank .

software engineering systems

♦ **Exists Contract Execution?**

♦ **Exists Contract Execution?**

▸ legal facts
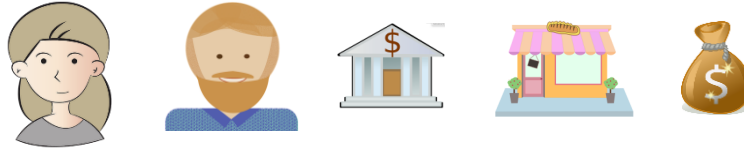
♦ **Exists Contract Execution?**

  ‣ legal facts

  ‣ legal claim set C

♦ **Exists Contract Execution?**
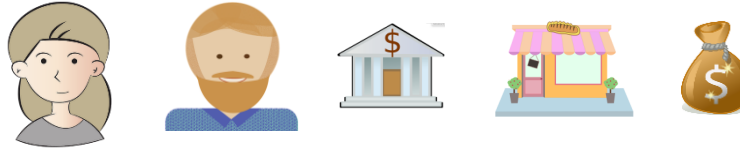
  ‣ legal facts

  ‣ legal claim set C

    – independent claim $T \in C_I$

    – consequence claim $WT \in C(T)$

3. Legal Encoding / Analysis
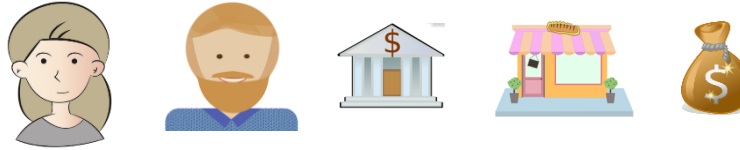
♦ **Exists Contract Execution?**

  ‣ legal facts

  ‣ legal claim set C

    – independent claim $T \in C_I$

    – consequence claim $WT \in C(T)$

      • e.g. $\phi_{WT} \equiv (d_{WT} = -1) \vee$ `T.DueDate` $< d_{WT}$

software engineering systems

3. Legal Encoding / Analysis

♦ **Exists Contract Execution?**

▸ legal facts

▸ legal claim set C

   – independent claim $T \in C_I$

   – consequence claim $WT \in C(T)$

      • e.g. $\phi_{WT} \equiv (d_{WT} = -1) \vee \texttt{T.DueDate} < d_{WT}$

▸ analysis

   – $\phi_{SPA} \equiv \phi_{owner} \; \bigwedge_{c \in C} \phi_c \; \bigwedge_{c \in C_I}(d_c \geq 0 \bigvee_{c' \in C(c)} d'_c \geq 0)$

software engineering systems

3. Legal Encoding / Analysis

♦ **Exists Contract Execution?**

▸ legal facts

▸ legal claim set C

   – independent claim $T \in C_I$

   – consequence claim $WT \in C(T)$

      • e.g. $\phi_{WT} \equiv (d_{WT} = -1) \lor$ `T.DueDate` $< d_{WT}$

▸ analysis

   – $\phi_{SPA} \equiv \phi_{owner} \; \bigwedge_{c \in C} \phi_c \; \bigwedge_{c \in C_I} (d_c \geq 0 \bigvee_{c' \in C(c)} d'_c \geq 0)$

software engineering systems

3. Legal Encoding / Analysis

♦ **Exists Contract Execution?**
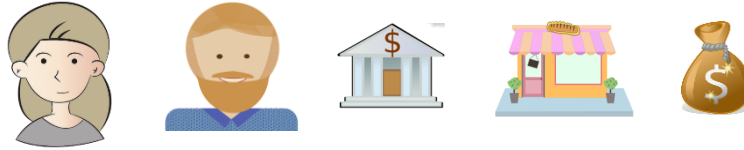
▸ legal facts

▸ legal claim set C

    – independent claim $T \in C_I$

    – consequence claim $WT \in C(T)$

        • e.g. $\phi_{WT} \equiv (d_{WT} = -1) \vee$ `T.DueDate` $< d_{WT}$

▸ analysis

    – $\phi_{SPA} \equiv \phi_{owner} \; \bigwedge_{c \in C} \phi_c \; \bigwedge_{c \in C_I}(d_c \geq 0 \bigvee_{c' \in C(c)} d'_c \geq 0)$

3. Legal Contract Analysis
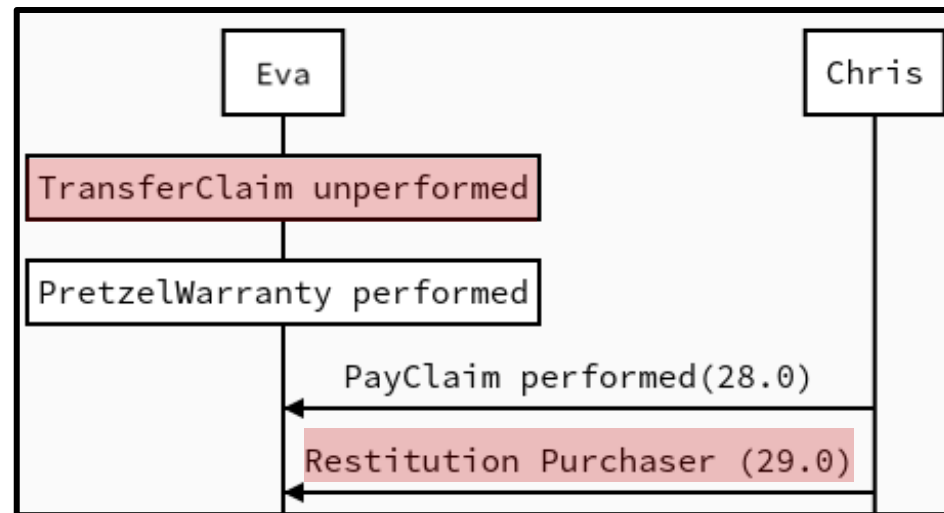
♦ **Limitation Time Analysis**

▸ claim limitation before due date

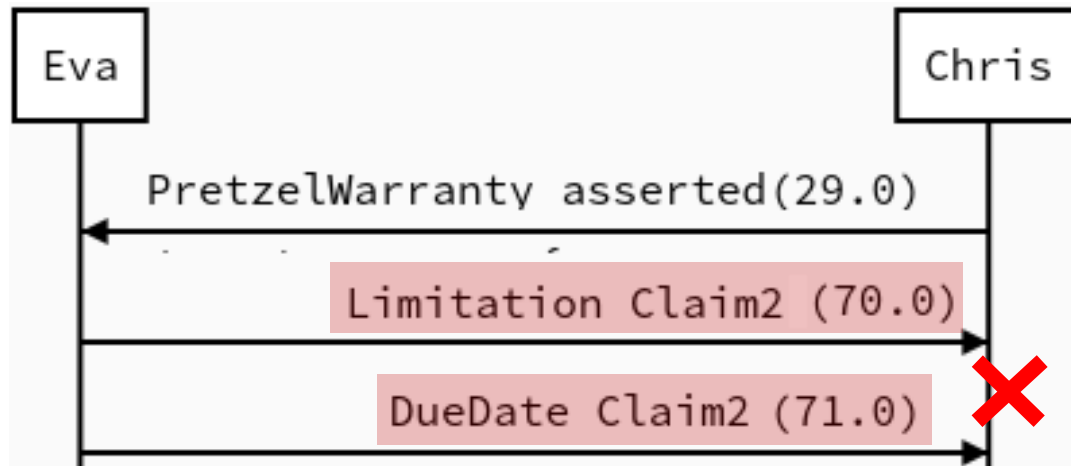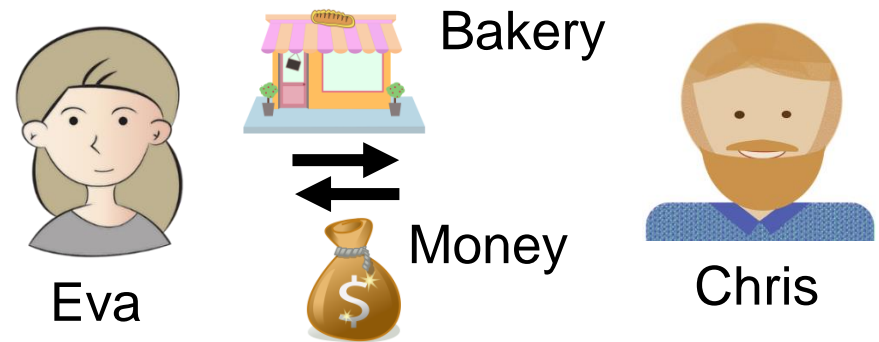# 3. Legal Contract Analysis

♦ **Limitation Time Analysis**

▸ claim limitation before due date

▸ e.g. analysis of compensation Claim2

$$- \phi_L \equiv \phi_{SPA} \wedge \texttt{Claim2.Limitation < Claim2.DueDate}$$

3. Legal Contract Analysis

♦ **Limitation Time Analysis**

▸ claim limitation before due date

▸ e.g. analysis of compensation Claim2

- $\phi_L \equiv \phi_{SPA} \land$ `Claim2.Limitation < Claim2.DueDate`

3. Legal Contract Analysis

♦ **Limitation Time Analysis**

▸ claim limitation before due date

▸ e.g. analysis of compensation Claim2

$-\ \phi_L \equiv \phi_{SPA} \wedge$ `Claim2.Limitation < Claim2.DueDate`

$\equiv \phi_{SPA} \wedge \boxed{70 < d_{Warranty} + 28 + 14}$

3. Legal Contract Analysis

♦ **Limitation Time Analysis**

‣ claim limitation before due date

‣ e.g. analysis of compensation Claim2

– $\phi_L \equiv \phi_{SPA} \land$ `Claim2.Limitation < Claim2.DueDate`
$\equiv \phi_{SPA} \land 70 < d_{Warranty} + 28 + 14$

3. Legal Contract Analysis

♦ **Limitation Time Analysis**

▸ claim limitation before due date

▸ e.g. analysis of compensation Claim2

$$- \phi_L \equiv \phi_{SPA} \wedge \texttt{Claim2.Limitation < Claim2.DueDate}$$
$$\equiv \phi_{SPA} \wedge 70 < d_{Warranty} + 28 + 14$$

Conclusion

♦ **Analysis of Legal Contracts**

- ‣ SPA
  - – encoding in SMT
  - – consistency analyses

Eva

Bakery

Money

Chris

software engineering systems

Conclusion

♦ **Analysis of Legal Contracts**
  ‣ SPA
    – encoding in SMT
    – consistency analyses
  ‣ tool LegalCheck

Eva   Bakery   Money   Chris

Blocks → encode → Object Diagram → formalize → SMT Tool Lib Code → Z3 → Results

software engineering systems

Conclusion

- **Analysis of Legal Contracts**
  - SPA
    - encoding in SMT
    - consistency analyses
  - tool LegalCheck

Blocks → encode → Object Diagram → formalize → SMT Tool Lib Code → Z3 → Results
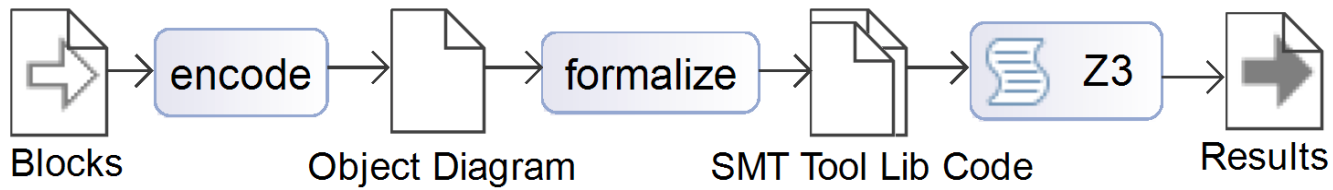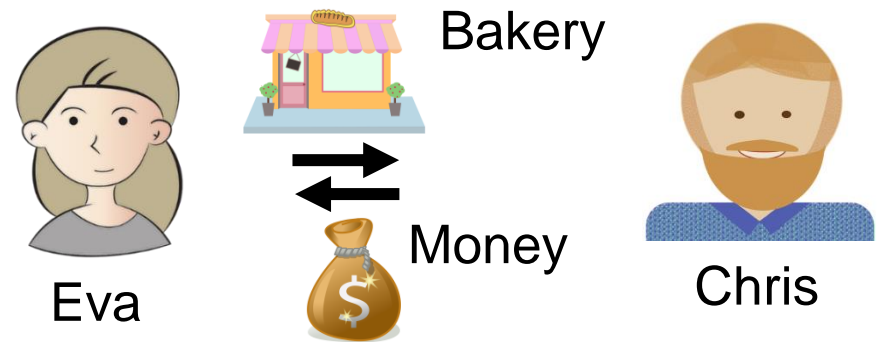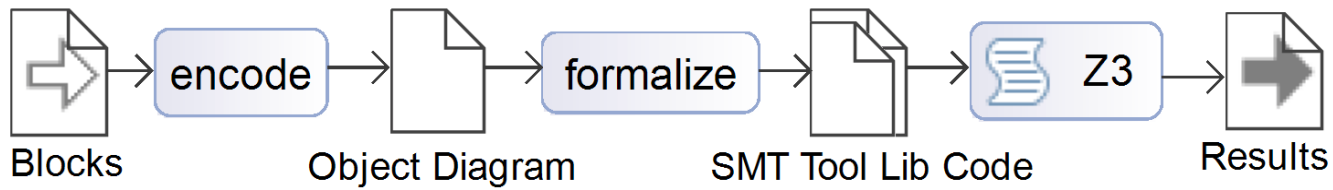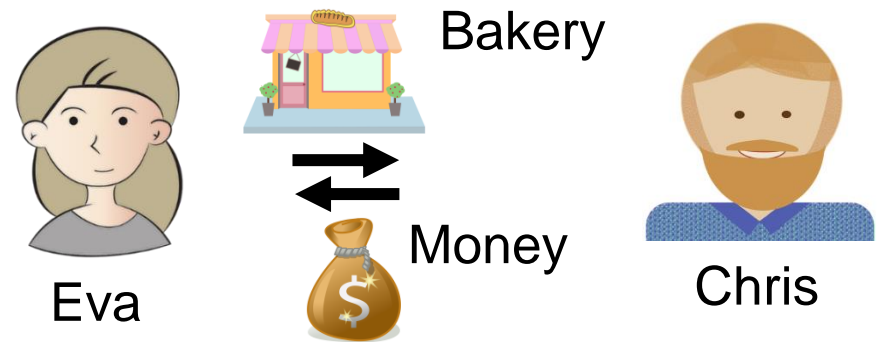
- **On-Going Work**
  - stepwise execution of contract
    - encode claim behavior
      - state machine
      - actions: arise, assert, perform, expire
  - analyze realistic size SPA
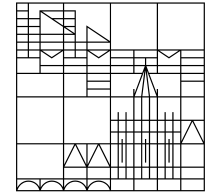
Universität
Konstanz

Thank you for your attention