Multiple viewpoints contracts for embedded systems

Albert Benveniste and Benoît Caillaud – INRIA Rennes
Roberto Passerone – PARADES / University of Trento

Foundations of Component Based dDesign, ESWEEK, 30 September 2007
Motivations and Contribution
Principles of component based design: interfaces + substituability in any context
Principles of component based design: interfaces + substitutuability in any context
Principles of component based design: splitting of responsibilities $\Rightarrow$ A/G reasoning
Principles of component based design: splitting of responsibilities $\Rightarrow$ A/G reasoning

(Assumption, Guarantee) : Contract
Embedded systems possess many components + different viewpoints

- Function
- Timing
- Reliability
- Energy
- QoS
- ...

- The designer may want to:
  - consider all viewpoints for each component
  - implement each component
  - compose the implementations

- Alternatively she may want to consider viewpoints incrementally:
  - consider all viewpoints for each component except Safety + QoS
  - implement each component
  - compose the implementations
  - Revisit her design for safety and QoS, possibly with a different, coarser grain, architecture
Embedded systems possess many components + different viewpoints

• Combining contracts for the different viewpoints of a same component ≠

• Combining contracts for different components

• The designer may want to:
  • consider all viewpoints for each component
  • implement each component
  • compose the implementations

• Alternatively she may want to consider viewpoints incrementally:
  • consider all viewpoints for each component except Safety + QoS
  • implement each component
  • compose the implementations
  • Revisit her design for safety and QoS, possibly with a different, coarser grain, architecture

• Is it a problem? Yes, for A/G reasoning…
Always: conjunction of Guarantees
Assumptions when combining components?

Assumptions on the considered contract can be (in part) discharged by the other component

[Dill, Negulescu, de Alfaro-Henzinger]
Always: conjunction of Guarantees
Assumptions when combining viewpoints?

Suppose that the different viewpoints do not interact; every implementation should satisfy all contracts the equivalent contract is l.u.b. for refinement
Always: conjunction of Guarantees
Combining Assumptions in general?

Assumptions on the considered contract can be (in part) discharged by the other component
[Dill, Negulescu, de Alfaro, Hänzinger]

Suppose that the different viewpoints do not interact; every implementation should satisfy all contracts the equivalent contract is l.u.b. for refinement

A blend of the above two is needed
Main contribution

- **Fusion** of contracts: new operation that subsumes the above two cases

- Supports:
  - Combining contracts for different components
  - Combining contracts for different (possibly interacting) viewpoints in a same component

- Supports:
  - Incremental design in both components and viewpoints
  - With consistent results in terms of possible implementations
Framework and Results
Contracts and Implementations

*\( M \) (implementations), *\( A \) (assumptions), *\( G \) (guarantees) are sets of runs composing by intersection and equipped with negation

**Contract:** \( C = (A,G) \)

**Implementation:** \( M \models C \iff M \cap A \subseteq G \)

ensures that \( M \) guarantees \( G \) in any context offering \( A \)

contracts having identical sets of implementations are said equivalent contract \( C \) in canonical form if:

\( G = MC \iff G \supseteq \neg A \)

\( \neg A \)
 Contracts and Implementations

\( M \) (implementations), \( A \) (assumptions), \( G \) (guarantees) are sets of runs composing by intersection and equipped with negation.

**Contract:** \( C = (A,G) \)

**Implementation:** \( M \models C \iff M \cap A \subseteq G \)

ensures that \( M \) guarantees \( G \) in any context offering \( A \)

Maximal implementation of \( C \): \( M_C = G \cup \neg A \)

contracts having identical sets of implementations are said equivalent

contract \( C \) in canonical form if: \( G = M_C \iff G \supseteq \neg A \)
Substituuality

Say that contract $C = (A,G)$ refines $C' = (A',G')$, written

$$C \leq C'$$

iff any implementation of $C$ is also an implementation of $C'$

$C \leq C'$ is ensured by

$$A \supseteq A' \text{ and } G \subseteq G'$$
Operations on Contracts (in canonical form): Combining Components [Dill, Negulescu, de Alfaro-Henzinger]

Let $C' = (A', G')$ and $C'' = (A'', G'')$ be two contracts associated with two interacting components.

$C = C' \parallel C''$ is the contract in which guarantees are composed (by conjunction) and assumptions are, in part, discharged by the other component:

$G = G' \cap G''$

$A = (A' \cap A'') \cup \neg (G' \cap G'')$
Operations on Contracts (in canonical form): Combining Viewpoints for 1 component

Let \( \wedge \) be the least upper bound for the dominance partial order \( \leq \)

Contract \( C \wedge C' \) subsumes the two contracts \( C \) and \( C' \)

\[
G = G' \cap G''
\]
\[
A = A' \cup A
\]
Operations on Contracts (in canonical form): Combining Viewpoints for 1 component

\[ M \models C \text{ and } M \models C' \iff M \models C \land C' \]

Hence, assuming \( C \) and \( C' \) do not interact:

\[ M \] satisfies the two contracts \( C \) and \( C' \)

\[ \iff \]

\[ M \models C \land C' \]

\[ G = G' \cap G'' \]

\[ A = A' \cup A \]
Operations on Contracts (in canonical form): General case??

Fusion of contracts
Operations on Contracts (in canonical form): Contract Fusion

\[ \text{fuse}[(C_i)_{i \in I}]_Q = \land_{J \subseteq I} [\parallel_{j \in J} C_j]_Q \]
A theorem regarding system design methodology

Suppose you have several viewpoints (function, QoS, safety…) to be addressed and you have several sub-systems or components.

What if

• You consider viewpoints incrementally for the entire system

  equivalent

• You consider all viewpoints for each component and then compose implementations

by special associativity rule for fusion
Discussion and conclusion

- A generic theory of contracts addressing the new problems raised by multiple viewpoints
- Good: the issue of multiple viewpoints in A/G reasoning is now solved
- But:
Discussion and conclusion

- A generic theory of contracts addressing the new problems raised by multiple viewpoints
- Good: the issue of multiple viewpoints in A/G reasoning is now solved
- But:
  - Dealing with assumptions as in the theory may not be user friendly (the user may not want to state « obvious » facts about what the environment should not do)
  - Making effective the operations $\cap$, $\cup$, $\exists Q.A$, $\neg$

? Investigating residuation as a substitute for $\neg$
THANK YOU