UNIVERSITY OF TWENTE.

AN IMPROVED LANGUAGE FOR HIGH LEVEL CONTROL FLOW SEMANTICS DEFINITION

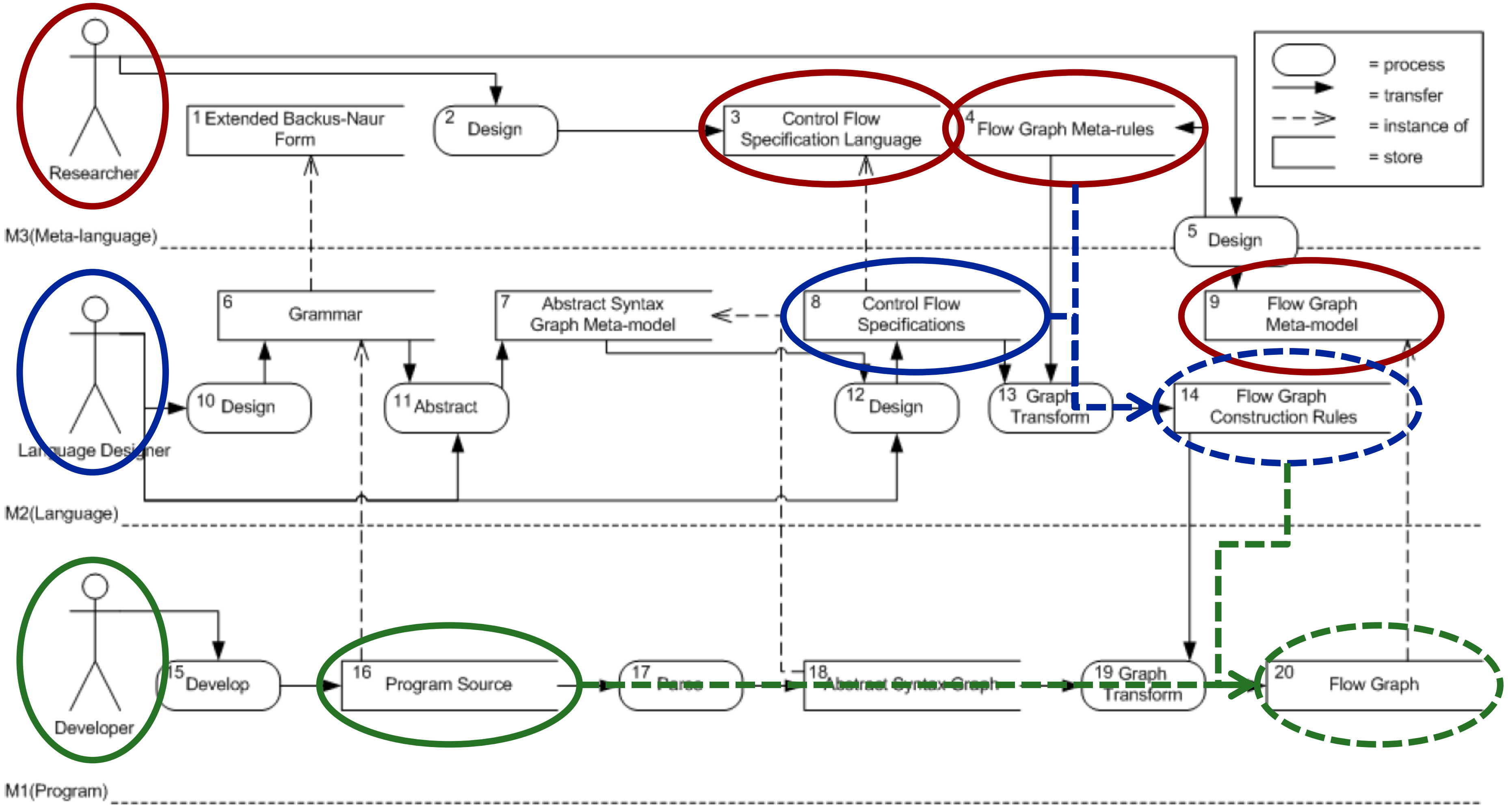
Richard Gankema, Arend Rensink, University of Twente

Graphs as Models, Eindhoven, April 2016

CONTEXT: SOFTWARE LANGUAGE DESIGN

- How do you *precisely* specify a software language?
  - Imagine Java
- Ingredients
  1. Syntax (grammar)
  2. Static semantics (scoping, typing, binding)
  3. Dynamic semantics (run-time)
- Observations
  1. Solved (EBNF, parser generators)
  2. Solved (but no standardised approach)
  3. Unsolved (hypothesis: graph transformation is a good approach)
- Here: sub-problem of 3 (semantics)
  - Control flow specification
  - Solved generically: control flow specification language
  - Operationalised by extracting control flow graph from syntax graph

ROLES AND ACTIVITIES

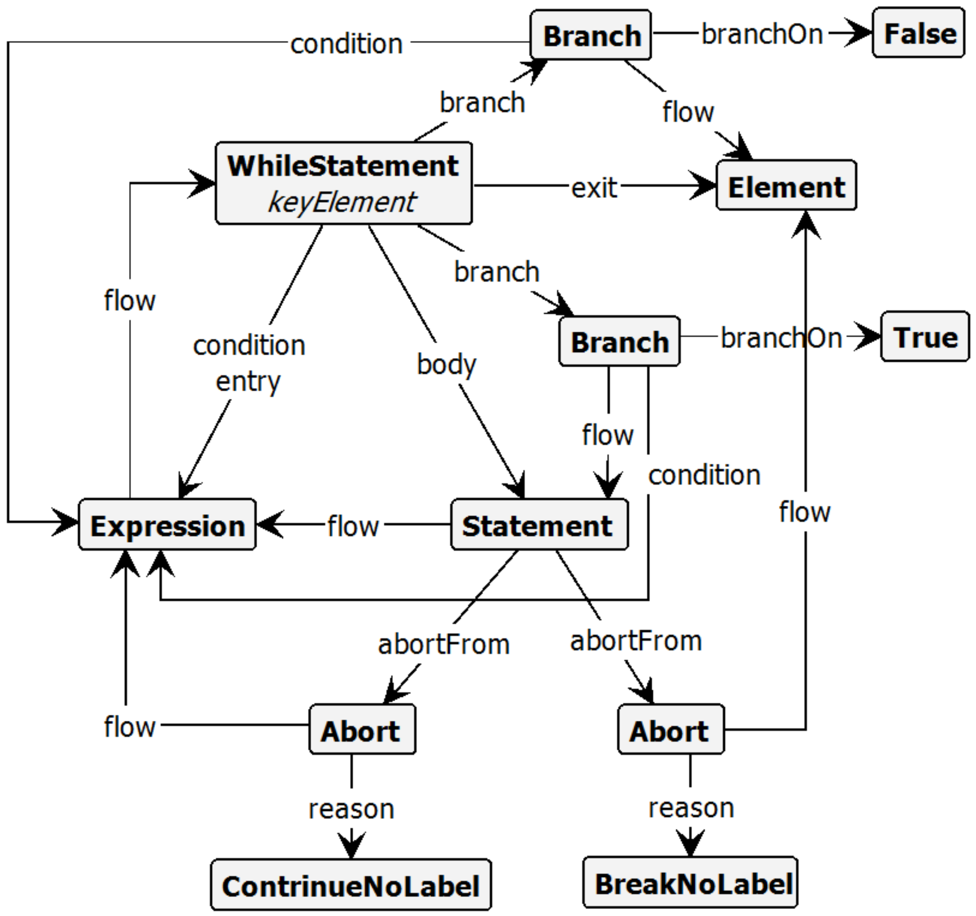


CONTRIBUTION OF THIS PAPER

- Design concrete syntax for CFSL
  - Readable & appealing
  - What are good (general) design principles for graphical languages?
- Provide tool support
  - Translation to abstract syntax

- Guiding principles
  - *The physics of notations: Toward a scientific basis for constructing visual notations in software engineering*, D. Moody, IEEE Transactions on Software Engineering, 2009.

CFSL – ABSTRACT SYNTAX



1. Bare AST
2. Basic flow
3. Nodified branches
4. Branch reasons
5. Branch conditions
6. Break statements
7. Continue statements

PHYSICS OF NOTATIONS

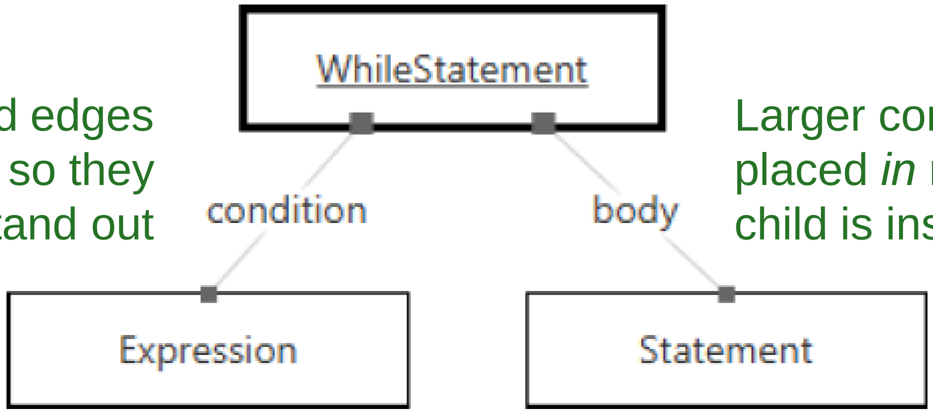
CFSL score

- Semiotic Clarity
  - One-to-one semantic constructs ↔ graphical symbols ✗
- Perceptual Discriminability
  - Different graphical symbols easily distinguishable ✓
- Semantic Transparency
  - Graphical symbols suggest their true meaning ✗
- Complexity Management
  - Explicit mechanisms to support complexity ✓/✗
- Visual Expressiveness
  - Use full range of visual variables ✗
- Dual Coding
  - Use text to support (rather than complement) graphics ✗
- Graphic economy
  - Number of different symbols coginitively manageable ✓

DESIGNING CFSL+

Key element underlined & bold outline

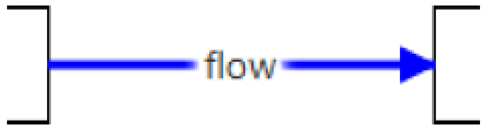
Child edges light gray so they do not stand out



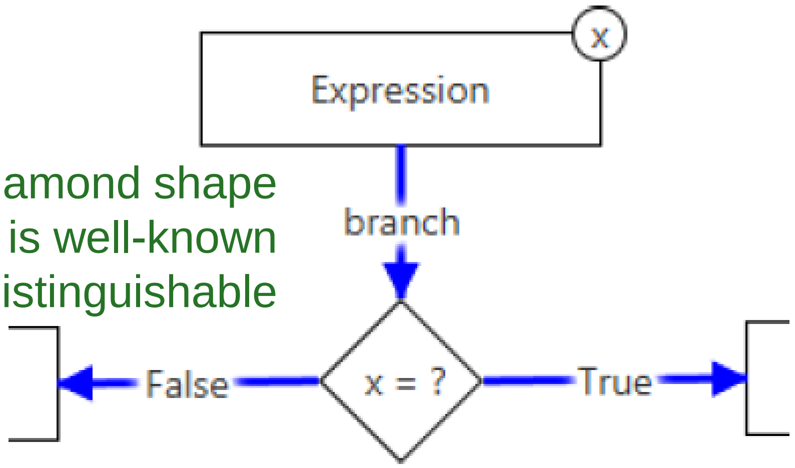
Larger connector on parent side placed *in* rectangle to suggest child is inside

Connectors are not arrows as there is no clear directionality

BASIC FLOW AND BRANCHING



Bold blue to appear in foreground;
arrow symbol is appropriate



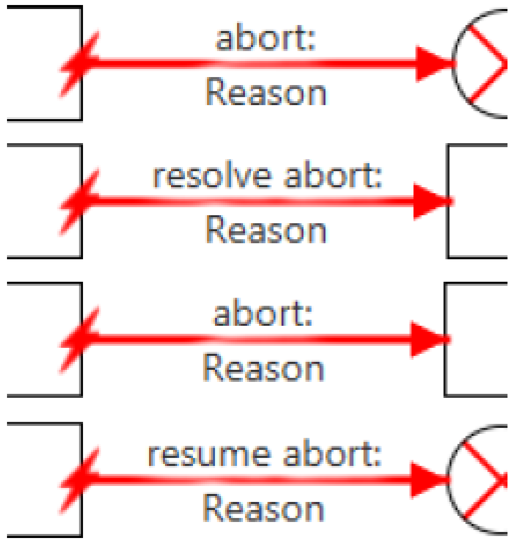
Diamond shape
because it is well-known
& distinguishable

Only name to refer to condition
for the sake of simplicity

ABRUPT FLOW

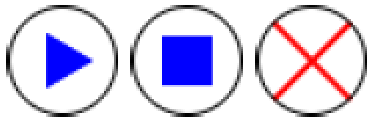
Red & thunderbolt
to indicate exceptional/
erroneous situation

Distinct from basic
control flow



SPECIAL NODES

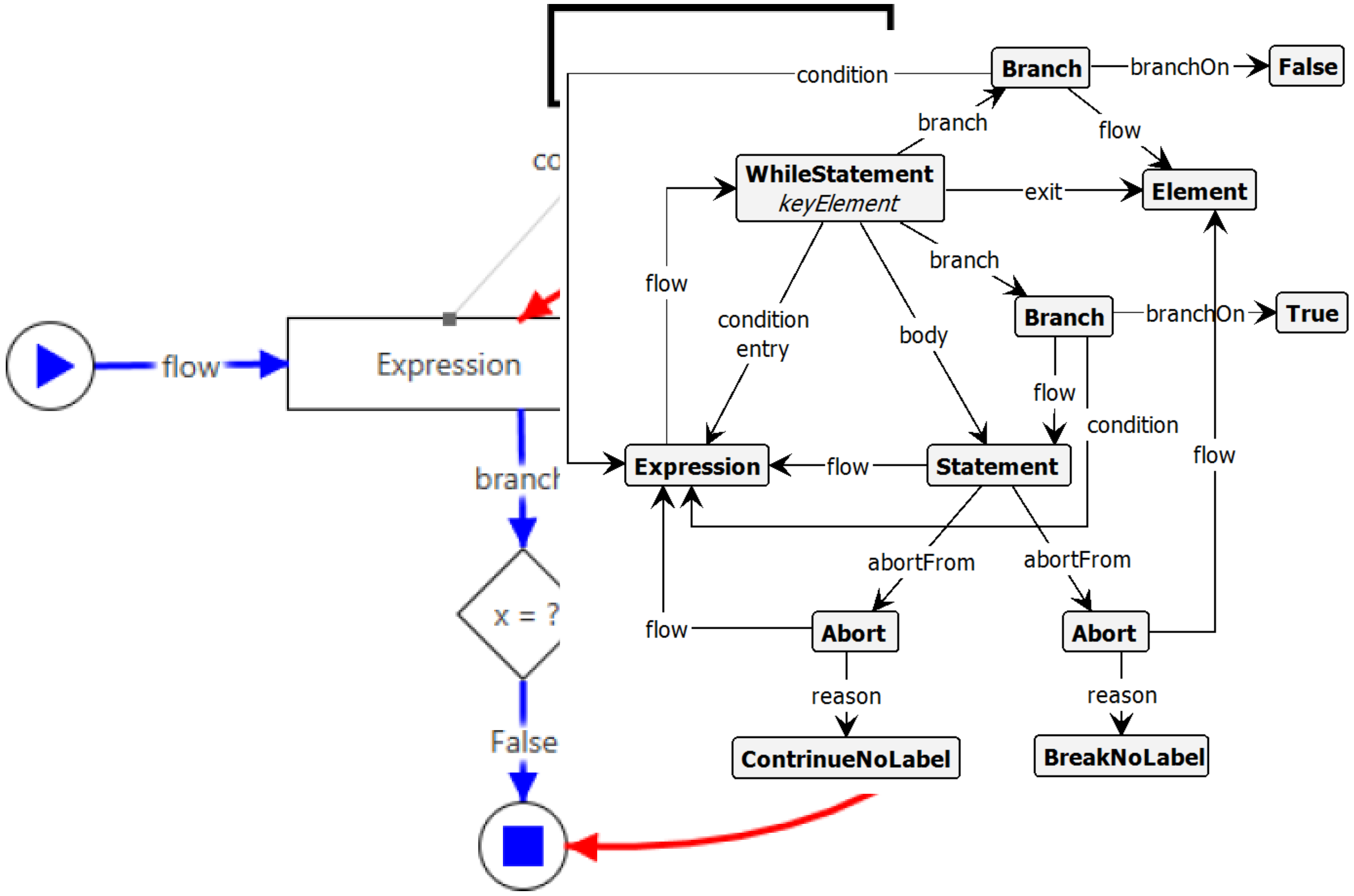
Start, stop, abort:
Common symbols



`Shape distinct from
syntax and branch nodes

Colour suggests link
to basic and abrupt flow

ALL TOGETHER NOW



EVALUATION

- Concrete syntax designed according to guidelines
- Implementation
  - Graphical editor
  - Translation to abstract syntax
- Planned user evaluation
  - Not yet carried out

PHYSICS OF NOTATIONS

	CFSL score	CFSL+ score
Semiotic Clarity	✗	✓
Perceptual Discriminability	✓	✓
Semantic Transparency	✗	✓
Complexity Management	✓/✗	✓/✗
Visual Expressiveness	✗	✓
Dual Coding	✗	✓
Graphic economy	✓	✓

More information and nuances in paper