Andrew Broad

PhD Oral Examination

18th September 2002

Comparative Code Understanding of Information Models

Code-Understanding System (CUS)
extracts Higher-Level Constraints (HLCs) from an EXPRESS model

Comparative Code-Understanding System (CCUS)
compares two EXPRESS models

Contributions:

- Inferring correspondences
- Assessing semantic equivalence and semantic strength
- Combining comparison and extraction
- Discriminative extraction and comparison

Experiment #17: Comparing the EDIF 3 0 0 and CFI DR Information Models

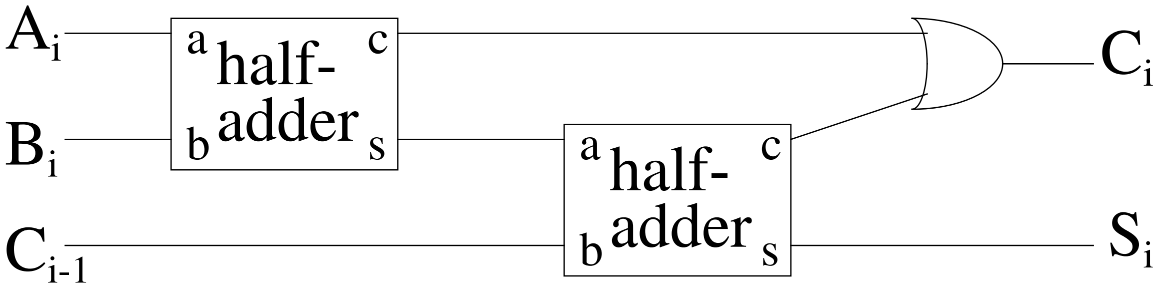
- EDIF 3 0 0:

  - Electronic Design Interchange Format version 3 0 0
  - Level 0 Information Model
  - A neutral format to exchange design-information among CAD/CAM tools
  - Connectivity (netlist) view
  - Schematic view

- CFI DR:

  - CAD Framework Initiative Design Representation
  - Limited to electrical connectivity, which is just part of EDIF
  - So we can compare the CFI DR model with the EDIF connectivity view

Entity-Correspondences



CFI DR model		EDIF 3 0 0 model
`cfidrLib`	↔	`library`
`cfidrCell`	↔	`cell`
`cfidrNetlistView`	↔	`cluster`
`cfidrInst`	↔	`instance`
`cfidrPortScalar`	↔	`master_port`
`cfidrPortBundle`	↔	`master_port_bundle`
`cfidrPortInstScalar`	↔	`instance_port`
`cfidrPortInstBundle`	↔	`instance_port_bundle`
`cfidrNetScalar`	↔	`signal`
`cfidrNetBundle`	↔	`signal_group`

Needed extensions to the CCUS:

- Suggest `cfidrNetlistView` and `cluster` as a possible correspondence - interact with user.
- How to suggest this correspondence:
  - structural similarity?
  - use an ontology?

Attribute-Correspondences
(**cfidrNetScalar/signal**)

```
ENTITY cfidrNetScalar SUBTYPE OF (cfidrNet);
  SELF\cfidrNet.Ports : SET [0:?] OF cfidrPortScalar;
  SELF\cfidrNet.PortInsts : SET [0:?] OF cfidrPortInstScalar;
  IsGlobal : cfidrBooleanT;
...
END_ENTITY;
```

```
ENTITY signal;
  joined_global_ports : OPTIONAL SET [1:?] OF global_port;
  joined_instance_member_ports :
                    OPTIONAL SET [1:?] OF instance_member_port;
  joined_master_ports : OPTIONAL SET [1:?] OF master_port;
  name : OPTIONAL name_information;
  properties : OPTIONAL SET [1:?] OF property;
...
END_ENTITY;
```

CFI DR model	EDIF 3 0 0 model
cfidrObject$ObjectId	SELF
cfidrObject$ObjectType	TYPEOF
cfidrObject$Props	properties
cfidrNamedObject$Name	name.primary_name.original_name.name
cfidrNet$NetBundles	signal_group$members
cfidrNet$Owner	DERIVE containing_view
cfidrNet$PortInsts	joined_instance_member_ports
cfidrNet$Ports	joined_master_ports

Needed extensions to the CCUS:

- Propagate attributes from supertypes to subtypes.
- Handle attributes which correspond to attributes in non-corresponding entities (`cfidrNet$NetBundles` and `signal_group$members`).

Attribute-Correspondences
(`cfidrPortScalar/master_port`)

```
TYPE cfidrPortDirection = ENUMERATION OF
                                (CFIDR_UNDEFINED_PORTDIRECTION,
                          CFIDR_INPUT, CFIDR_OUTPUT, CFIDR_IO);
END_TYPE;

ENTITY cfidrPortScalar SUBTYPE OF (cfidrPort);
  NetScalar : OPTIONAL cfidrNetScalar;
  Direction : cfidrPortDirection;
WHERE
  r1: NOT EXISTS(NetScalar) OR (SELF IN NetScalar.Ports);
END_ENTITY;
```

```
ENTITY master_port ABSTRACT SUPERTYPE OF
                 (ONEOF (input_master_port, output_master_port,
bidirectional_master_port, unspecified_direction_master_port));
  ...
  name : OPTIONAL name_information;
  ...
  unused_internally : boolean_token;
...
END_ENTITY;
```

CFI DR model	EDIF 3 0 0 model
`cfidrNamedObject$Name`	`name.primary_name.original_name.name`
`NOT EXISTS(NetScalar)`	`unused_internally`
`Direction`	`ABSTRACT SUPERTYPE OF (...)`
`.CFIDR_UNDEFINED_PORTDIRECTION.`	`$unspecified_direction_master_port$`
`.CFIDR_INPUT.`	`$input_master_port$`
`.CFIDR_OUTPUT.`	`$output_master_port$`
`.CFIDR_IO.`	`$bidirectional_master_port$`

Needed extensions to the CCUS:

- Propagate attributes from supertypes to subtypes;
- Handle encapsulated entitities
  (`Name` corresponds to `name.primary_name...`);
- Attributes may correspond to Boolean-valued expressions!

Experimental Results for EDIF 3 0 0 and CFI DR

659 HLCs extracted from first model (EDIF 3 0 0 Level 0)
61 HLCs extracted from second model (CFI DR)

Maximum theoretical number of HLC-comparisons:
659 * 61 = 40199

Mode	E	-T	-U	-S	-H	-R	M	X
HLCs 1st model	659	659	659	659	0	659	659	659
HLCs 2nd model	61	63	61	61	0	59	59	61
Comparisons	109117	106146	97804	92838	1952	8108	69432	109117
HLC-comparisons	14667	13540	14667	14647	0	655	5005	14667
% max. HLC-cmps.	36%	34%	36	36%	0%	2%	12%	36%
Correspondences	136	52	136	109	34	52	81	136
HLC-corrs.	26	8	26	25	0	8	8	26
Differences	12311	12058	11924	11622	188	753	6665	12311
FC-thes. lookups	52735	0	42977	42485	4	2581	40915	52735
Name-thes. lookups	41449	0	41029	34168	1478	4698	16430	41449

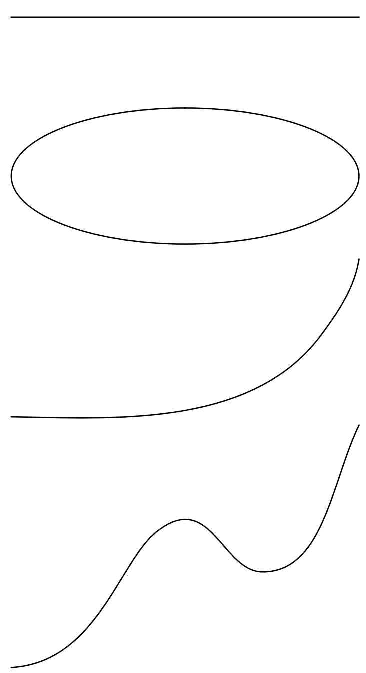
The Modes:
- E: all features enabled
- -T: thesauri disabled
- -U: subexpression cases disabled
- -S: SE/S cases disabled
- -H: extraction of HLCs disabled
- -R: no remembrance
- M: the monogamy remembrance heuristic
- X: submitting extraction prompts together

Experiment #18: Comparing the Gerber Conceptual Model with the SVG Model

- Scalable Vector Graphics (SVG):
  - A language for graphics
  - SVG Model 4.7

- Gerber:
  - A language for photoplotting
  - 3 EXPRESS models of Gerber:
    - + Conceptual Model
    - + Implementation Model
    - + Syntax Model
  - Conceptual Model is closest to SVG Model

Entity-Correspondences



Gerber Conceptual Model		SVG Model
gerber_plot	↔	outMostSvgElement
coordinate	↔	coordinate
gerber_point	↔	point
gerber_point	↔	Point3D
linear_interpolation	↔	line
circular_interpolation	↔	ellipticalArcCurve
parabolic_interpolation	↔	quadraticBezierSegment
cubic_interpolation	↔	cubicBezierSegment
symbol	↔	symbol

Needed extensions to the CCUS:

- Veto homonyms!

Attribute-Correspondences (gerber_plot/outMostSvgElement)

```
ENTITY gerber_plot;
  image_name : character_string;
  image_polarity : polarity;
  image_rotation : rotation;
  plotter_destination : plotter;
  plot_film_type : film_type;
  image_x_justification : justification;
  image_y_justification : justification;
  image_layers : SET OF image_layer;
END_ENTITY;
```

```
ENTITY outMostSvgElement SUBTYPE OF (svgelement);
  SELF\container.etitle : title;
  SELF\svgElement.zoomAndPan : BOOLEAN;
END_ENTITY;
```

Gerber Conceptual Model		SVG Model
image_name	↔	SELF\container.etitle.eContent
image_x_justification	↔	svgElement$viewPort.topLeftPoint.x

Needed extensions to the CCUS:

- Compare attributes with attribute-redeclarations.
- Propagate attributes from supertypes to subtypes.
- Infer correspondence between an attribute and an attribute in an encapsulated entity.
- Take (partial) specification of the mapping as input.

Attribute-Correspondences
(`coordinate/coordinate`)

```
ENTITY coordinate ABSTRACT SUPERTYPE OF
        (ONEOF (negative_coordinate, non_negative_coordinate));
  coordinate_value : distance;
  measuring_unit : gerber_unit;
...
END_ENTITY;

TYPE gerber_unit = ENUMERATION OF (inch, millimeter);
END_TYPE;
```

```
ENTITY coordinate ABSTRACT SUPERTYPE OF
              (ONEOF (absoluteCoordinate, relativeCoordinate));
  coordinateValue : svgLength;
END_ENTITY;

ENTITY svgLength ABSTRACT SUPERTYPE OF
                      (ONEOF (... SvgLengthInMillimetre, ...));
    lengthValueInPixel : valueSpecification;
END_ENTITY;
```

Gerber Conceptual Model	SVG Model
`coordinate_value`	`coordinateValue.lengthValueInPixel`
`measuring_unit`	`TYPEOF(coordinateValue)`
`.inch.`	`$svgLengthInInch$`
`.millimeter.`	`$svgLengthInMillimetre$`

Needed extensions to the CCUS:

- Handle encapsulated entitities properly.
- Propagate `choice_set` on `svgLength` to `coordinate`.

Attribute-Correspondences

(`circular_interpolation/ellipticalArcCurve`)

```
ENTITY circular_interpolation ABSTRACT SUPERTYPE OF (...)
                              SUBTYPE OF (interpolation);
  start_point : gerber_point;
  end_point : gerber_point;
  center_point : basic_gerber_point;
END_ENTITY;
```

```
ENTITY ellipticalArcCurve SUBTYPE OF (pathdata);
  startPoint : point;
  drawToPoint : point;
  radiusX :svgLength;
  radiusY :svgLength;
  xAxisRotation : angle;
...
END_ENTITY;
```

Gerber Conceptual Model		SVG Model
start_point	↔	startPoint
end_point	↔	drawToPoint
center_point	↔	startPoint, drawToPoint, radiusX, radiusY, xAxisRotation

Needed extensions to the CCUS:

- Handle computational mappings.
- Handle one-to-many attribute-relationships.

Attribute-Correspondences

(`parabolic_interpolation`/`quadraticBezierSegment`)

```
ENTITY parabolic_interpolation ABSTRACT SUPERTYPE OF
                   (ONEOF (fixed_width_parabolic_interpolation,
                              tapered_parabolic_interpolation))
                               SUBTYPE OF (interpolation);
  through_points : LIST [3 : 3] OF gerber_point;
END_ENTITY;
```

```
ENTITY quadraticBezierSegment SUBTYPE OF (pathdata);
  controlledParameters : LIST [2:2] OF controlledPoint;
DERIVE
  SELF\pathData.head : point
    := controlledParameters[1].actualPoint;
  SELF\pathData.tail : point
    := controlledParameters[2].actualPoint;
  SELF\pathData.rendered : BOOLEAN
    := NOT(valueSame(controlledParameters));
WHERE
  oneControlPoint : controlledparameters[1].controlPoint =
                    controlledparameters[2].controlPoint;
END_ENTITY;
```

Gerber Conceptual Model	SVG Model
`through_points` ↔	`controlledParameters`

Needed extensions to the CCUS:

- It can infer this correspondence with the aid of a name-thesaurus, but it cannot understand it as I do!

Experimental Results for Gerber and SVG

90 HLCs extracted from first model (Gerber Conceptual)
135 HLCs extracted from second model (SVG)

Maximum theoretical number of HLC-comparisons:
90 * 135 = 12150

Mode	E	-T	-U	-S	-H	-R	M	X
HLCs 1st model	92	92	92	92	0	92	92	92
HLCs 2nd model	135	135	135	135	0	135	135	135
Comparisons	90755	71909	90401	75003	18844	31276	31976	90755
HLC-comparisons	6742	6742	6742	6742	0	1688	1732	6742
% max. HLC-cmps.	55%	55%	55%	55%	0%	14%	14%	55%
Correspondences	33	6	33	31	27	30	33	33
HLC-corrs.	0	0	0	0	0	0	0	0
Differences	3541	3451	3523	3522	86	974	1036	3541
FC-thes. lookups	43144	0	42891	31298	2	5212	5648	43144
Name-thes. lookups	79343	0	79339	64040	18721	28782	28808	79343

The Modes:
- E: all features enabled
- -T: thesauri disabled
- -U: subexpression cases disabled
- -S: SE/S cases disabled
- -H: extraction of HLCs disabled
- -R: no remembrance
- M: the monogamy remembrance heuristic
- X: submitting extraction prompts together

Experiment #19: Mr. Jones's Garden flattened up

- Mr. Jones's Garden:

  - A toy model from the MINT Group

  - Simple Solution

- Flatten-Up Transformation:

  - Flattening converts subtype hierarchies to flat entities

  - *Flatten-Up*: remove subtypes,
    moving attributes and constraints up to supertype

  - *Flatten-Down*: remove supertype,
    moving attributes and constraints down to subtypes

  - Mr Jones's Garden: Simple Solution flattened up

Entity-Correspondences

Mr. Jones's Garden	flattened up
`garden` ⟷	`garden`
`greenhouse` ⟷	`greenhouse`
`bed` ⟷	`bed`
`plant` ⟷	`flattened_up_plant`
`greenhouse_plant` ⟷	`.plant_greenhouse_plant.`
`outdoors_plant` ⟷	`.outdoors_plant_plant.`

Attribute-Correspondences (`plant/flattened_up_plant`)

```
ENTITY plant ABSTRACT SUPERTYPE OF
                                 (ONEOF (greenhouse_plant, outdoors_plant));
  colour : flower_colour;
...
END_ENTITY;

ENTITY greenhouse_plant SUBTYPE OF (plant);
  min_temp : temperature;
  max_temp : temperature;
WHERE
  the_temperatures_must_be_sensible: max_temp >= min_temp;
END_ENTITY;
```

```
ENTITY flattened_up_plant;
  instance_of : instances_in_flattened_up_plant;
  colour : flower_colour;
  min_temp : OPTIONAL temperature;
  max_temp : OPTIONAL temperature;
WHERE
  the_temperatures_must_be_sensible_from_greenhouse_plant:
    ((greenhouse_plant IN entities_in_flattened_up_instance
      (instance_of, combination_sets_for_flattened_up_plant))
      AND (max_temp >= min_temp))
    XOR NOT (greenhouse_plant IN entities_in_flattened_up_instance
             (instance_of, combination_sets_for_flattened_up_plant));

  mandatory_exists__of_greenhouse_plant:
    (EXISTS (min_temp) AND EXISTS (max_temp))
    XOR NOT (greenhouse_plant IN entities_in_flattened_up_plant_instance
             (instance_of, combination_sets_for_flattened_up_plant));
...
END_ENTITY;
```

Mr. Jones's Garden		flattened up
plant$colour	⟷	colour
greenhouse_plant$min_temp	⟷	min_temp
greenhouse_plant$max_temp	⟷	max_temp

Needed extensions to the CCUS:

- Extract HLCs from the subtype-attributes, and from the domain-rules, in order to assess the SE/S of their constraints.

Attribute-Correspondences (`plant/flattened_up_plant`)

```
ENTITY greenhouse;
  temp : temperature;
  holds_plants : SET [1:?] OF greenhouse_plant;
WHERE
  the_plants_can_survive_the_temperature:
    QUERY (p <* holds_plants | {p.min <= temp <= p.max_temp})
    = holds_plants;
END_ENTITY;
```

```
ENTITY greenhouse;
  temp : temperature;
  holds_plants : SET [1:?] OF flattened_up_plant;
WHERE
  the_plants_can_survive_the_temperature:
    QUERY (p <* holds_plants | {p.min <= temp <= p.max_temp})
    = holds_plants;

  attribute_reference_check_for_holds_plants:
    SIZEOF (QUERY (inst <* holds_plants |
            ((EXISTS(inst) AND NOT (greenhouse_plant IN
             entities_in_flattened_up_plant_instance
             (inst.instance_of,
             inst.combination_sets_for_flattened_up_plant))))))
    = 0;
END_ENTITY;
```

Mr. Jones's Garden		flattened up
`temp`	⟷	`temp`
`holds_plants`	⟷	`holds_plants`

Needed extensions to the CUS:

- A new HLC-class `subtype_reference`, extractable from:
  - a reference to a subtype-entity;
  - a domain rule of the form above (`attribute_reference_check_for_holds_plants`).
- Understanding user-defined functions.

Further Experiments since I submitted my Thesis

13. AP232

14. MPhil/PhD Frame-Class Models

15. Gerber: Syntax Model/Implementation Model

16. Gerber: Implementation Model/Conceptual Model

17. EDIF 3 0 0 / CFI DR

18. Gerber/SVG

19. Mr. Jones's Garden flattened up

Summarised CCUS Output

- HTML format
- organised around entity-pairs
- pruned down

Bug-Fixes

- repeated-comparison bugs
- abstract correspondences
- disabling SE/S cases

The data in Section 7.2.4 and Appendix F have been updated!