Verdi³ NPI Training
Text Model

Based on Verdi³ 2013.01

Glossary

- VIA = Verdi Interoperability Applications
- NPI = Novas Programming Interface
- Novas KDB = Novas Knowledge Database
- FSDB = Fast Signal Database
- VPI = Verilog Procedural Interface

Overview

- Text Model Introduction
- NPI Commands
- NPI Object Diagrams
- Case Study

Overview

- Text Model Introduction
- NPI Commands
- NPI Object Diagrams
- Case Study

DM Model Introduction

What is NPI Text Model?

- Lets users traverse and manipulate design source code from a text perspective
  - Every file, line, and word is regarded as an object
- Provides basic insert, delete, and replace capabilities
- Users can access the attribute of each word from the design analysis results of the Verdi system

Overview

- Text Model Introduction
- NPI Commands
- NPI Object Diagrams
- Case Study

NPI Commands

Get Handles

- **npi_text_file_by_name –name** *Filename*
  - Obtain a file handle with the specified full file name or local file name

- **npi_text_handle –type** *TextObjType* **–ref** *ReferenceHandle*
  - Obtain a handle that contains the reference handle object
    - Available values of *TextObjType* are **npiTextFile** and **npiTextLine**

- Example:

Get the handle for *example.v* file

```
% set file_hdl [ npi_text_file_by_name –name “example.v” ]

% set line_iter [ npi_text_iter_start –type “npiTextLine” –ref $file_hdl ]
% set line_hdl [ npi_text_iter_next –iter $line_iter ]

% set line_file_handle [ npi_text_handle –type “npiTextFile” –ref $line_hdl ]
```

Get the handle for the file that the line (*$line_hdl*) belongs to

NPI Commands

Create the Iterator and Iterate Items (1/2)

- **npi_text_iter_start –type** *TextObjType* **–ref** *ReferenceHandle*
  - Obtain an iterator to iterate the reference handle object
    - Available values of *TextObjType* are **npiTextFile**, **npiTextLine**, and **npiTextWord**
    - Specify an empty string (“”) to *ReferenceHandle* if the to-be iterated type is **npiTextFile**

- **npi_text_iter_next –iter** *IteratorHandle*
  - Iterate the iterator to the next element

- **npi_text_iter_stop –iter** *IteratorHandle*
  - Free the iterator, and release the occupied resource

NPI Commands

Create the Iterator and Iterate Items (2/2)

- Example:

```
set file_iter [ npi_text_iter_start –type “npiTextFile” –ref “” ]
```

Create an iterator *$file_iter* for all imported files

```
while { “” != [ set file_hdl [ npi_text_iter_next –iter $file_iter ] ] } {
```

Scan the iterator *$file_iter* and get the handle of the next file

```
  set file_name [ npi_text_property_str –type “npiTextFileName” –ref $file_hdl ]
  puts $file_name
}
npi_text_iter_stop –iter $file_iter
```

Free the iterator *$file_iter*

NPI Commands

Get Properties from Handles

- **npi_text_property –type** *TextPropertyType* **–ref** *TargetHandle*
  - Get the specified integer property of the text object

- **npi_text_property_str –type** *TextPropertyType* **–ref** *TargetHandle*
  - Get the specified string property of the text object

- Example:

```
% set file_hdl [ npi_text_file_by_name –name “example.v” ]
% puts [ npi_text_property –type “npiTextLineCount” –ref $file_hdl ]
% puts [ npi_text_property_str –type “npiTextFileName” –ref $file_hdl ]
```

Get the line count number of the file handle *$file_hdl*

Get the file name of the file handle *$file_hdl*

NPI Commands

Manipulate Lines (1/2)

- **npi_text_insert_line_before –ref** *ReferenceLineHandle* **–content** "*NewLine*"
  - Insert a line just before a specified line handle
- **npi_text_insert_line_after –ref** *ReferenceLineHandle* **–content** *"NewLine"*
  - Insert a line just after a specified line handle
- **npi_text_delete_line –ref** *TargetLineHandle*
  - Delete a line handle
- **npi_text_replace_line –ref** *TargetLineHandle* **–content** *"NewContent"*
  - Replace the content of the target line with the specified new content
- **NOTE1**: The *NewLine* and *NewContent* should be enclosed by double quotation marks, and they could be an empty string (“”)
- **NOTE2**: a change line (\n) will automatically be added to the end of the *NewLine*. If the *NewLine* argument is an empty string, this function still inserts a change line (\n) just before the reference line handle.

NPI Commands

Manipulate Lines (2/2)

- Example:

```
% set file_hdl [ npi_text_file_by_name –name “example.v” ]
% set line_iter [ npi_text_iter_start –type “npiTextLine” –ref $file_hdl ]
% set line_hdl [ npi_text_iter_next –iter $line_iter ]
% set inserted_line_hdl_before [ npi_text_insert_line_before –ref $line_hdl
–content “/* comment */” ]
```

Insert a line before the line handle *$line_hdl*

```
% set inserted_line_hdl_after [ npi_text_insert_line_after –ref $line_hdl –
content “  wire c;” ]
```

Insert a line before the line handle *$line_hdl*

```
% set replaced_line [ npi_text_replace_line –ref $line_hdl –content “/* port
declaration */” ]

% set delete_result [ npi_text_delete_line -ref $line_hdl ]
```

NPI Commands

Manipulate Words (1/2)

- **npi_text_insert_word_before –ref** *ReferenceWordHandle* **–word** *NewWord*
  - Insert a word just before the specified word handle
- **npi_text_insert_word_after –ref** *ReferenceWordHandle* **–word** *NewWord*
  - Insert a word just after the specified word handle
- **npi_text_replace_word –ref** *TargetWordHandle* **–word** *NewWord*
  - Replace the word of a specified word handle
- **npi_text_delete_word –ref** *TargetWordHandle*
  - Delete a word handle
- **NOTE**: The *NewWord* could be an empty string (“”)

NPI Commands

Manipulate Words (2/2)

- Example:

```
% set file_hdl [ npi_text_file_by_name –name “example.v” ]
% set line_iter [ npi_text_iter_start –type “npiTextLine” –ref $file_hdl ]
% set line_hdl [ npi_text_iter_next –iter $line_iter ]
% set word_iter [ npi_text_iter_start –type “npiTextWord” –ref $line_hdl ]
% set inserted_before_word_hdl [ npi_text_insert_word_before –ref $word_hdl –word “//” ]
```

Insert a word before the word handle *$word_hdl*

```
% set inserted_after_word_hdl [ npi_text_insert_word_after –ref $word_hdl –word “ //comment” ]
```

Insert a word after the word handle *$word_hdl*

```
% set replaced_word [ npi_text_replace_word –ref $word_hdl –word “AND_g1” ]
```

Replace the word for the word handle *$word_hdl*

```
% set delete_word [ npi_text_delete_word -ref $word_hdl ]
```

Delete the word handle *$word_hdl*

NPI Commands

Expand the Include File

- **npi_text_expand_include –ref** *ReferenceWordHandle*
  - Expand the content of the include file to a file

- Example:

```
set file_hdl [ npi_text_file_by_name –name “expand_include.v” ]
set line_iter [ npi_text_iter_start –type “npiTextLine” –ref $file_hdl ]
set line_hdl [ npi_text_iter_next –iter $line_iter ]
set word_iter [ npi_text_iter_start –type “npiTextWord” –ref $line_hdl ]
while { “” != [ set word_hdl [ npi_text_iter_next –iter $word_iter ] ] } {
 if { “npiTextIncludeFile” == [ npi_text_property_str –type
“npiTextWordAttribute” –ref $word_hdl ] } {
```

Check whether the property of the word handle is “include”

```
 set expand_flag [ npi_text_expand_include –ref $word_hdl ]
}
}
```

Expand the include file of the word handle

NPI Commands

Expand the Macro

- **npi_text_expand_macro –ref** *ReferenceWordHandle*
  - Expand macro description to a file

- Example:

```
set file_hdl [ npi_text_file_by_name –name “expand_include.v” ]
set line_iter [ npi_text_iter_start –type “npiTextLine” –ref $file_hdl ]
set line_hdl [ npi_text_iter_next –iter $line_iter ]
set word_iter [ npi_text_iter_start –type “npiTextWord” –ref $line_hdl ]
while { “” != [ set word_hdl [ npi_text_iter_next –iter $word_iter ] ] } {
  if { “npiTextMacroName” == [ npi_text_property_str –type
“npiTextWordAttribute” –ref $word_hdl ] } {
```

Check whether the property of the word handle is “macro”

```
  set expand_flag [ npi_text_expand_macro –ref $word_hdl ]
}
}
```

Expand the macro description of the word handle

Overview

- DM Model Introduction
- NPI Commands
- NPI Object Diagrams
- Case Study

How to Use Object Diagrams?

Guidelines for Reading Object Diagrams

- Which object diagram should I refer to?
  - Find the object diagram based on your reference object handle
  - For example, if you are looking for all words for a line, you need to check the "**line**" object diagram

- Which NPI command should I use to get the handle?
  - Refer to the object diagram according to your reference object handle, and check the arrow type
    -  Single arrow (one to one):

      npi_text_handle
    -  Double arrow (one to many):

      npi_text_iter_start

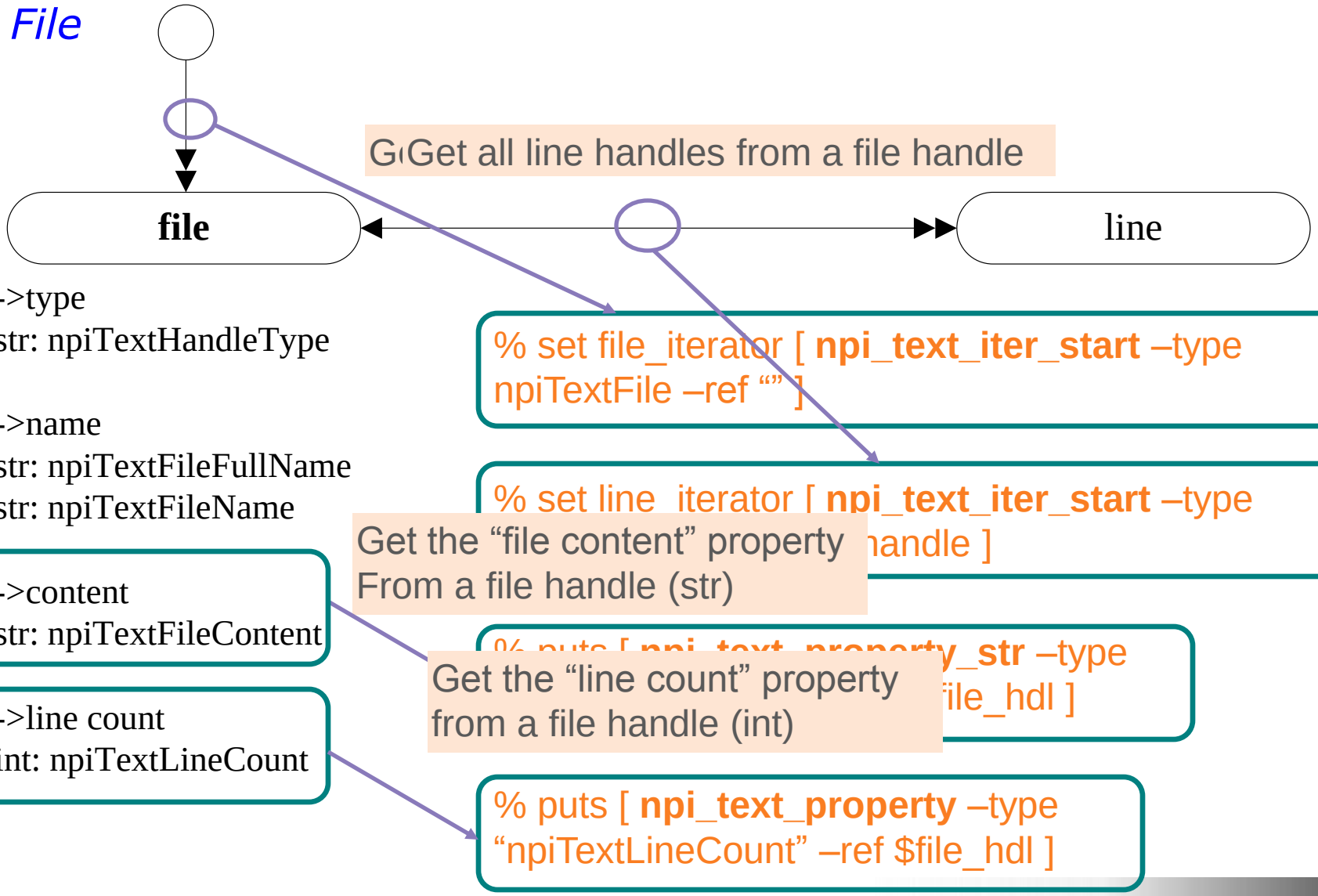
How to Use Object Diagrams?

Guidelines for Reading Object Diagrams

- How can I get the type string of **–type** option for **npi_text_handle** and **npi_text_iter_start**?
  - Refer to the object diagram according to your reference object handle, and check the name for each object
    - Change the first character to uppercase, and then add "**npiText**" as the prefix.
    - Available types are: **npiTextFile**, **npiTextLine**, and **npiTextWord**

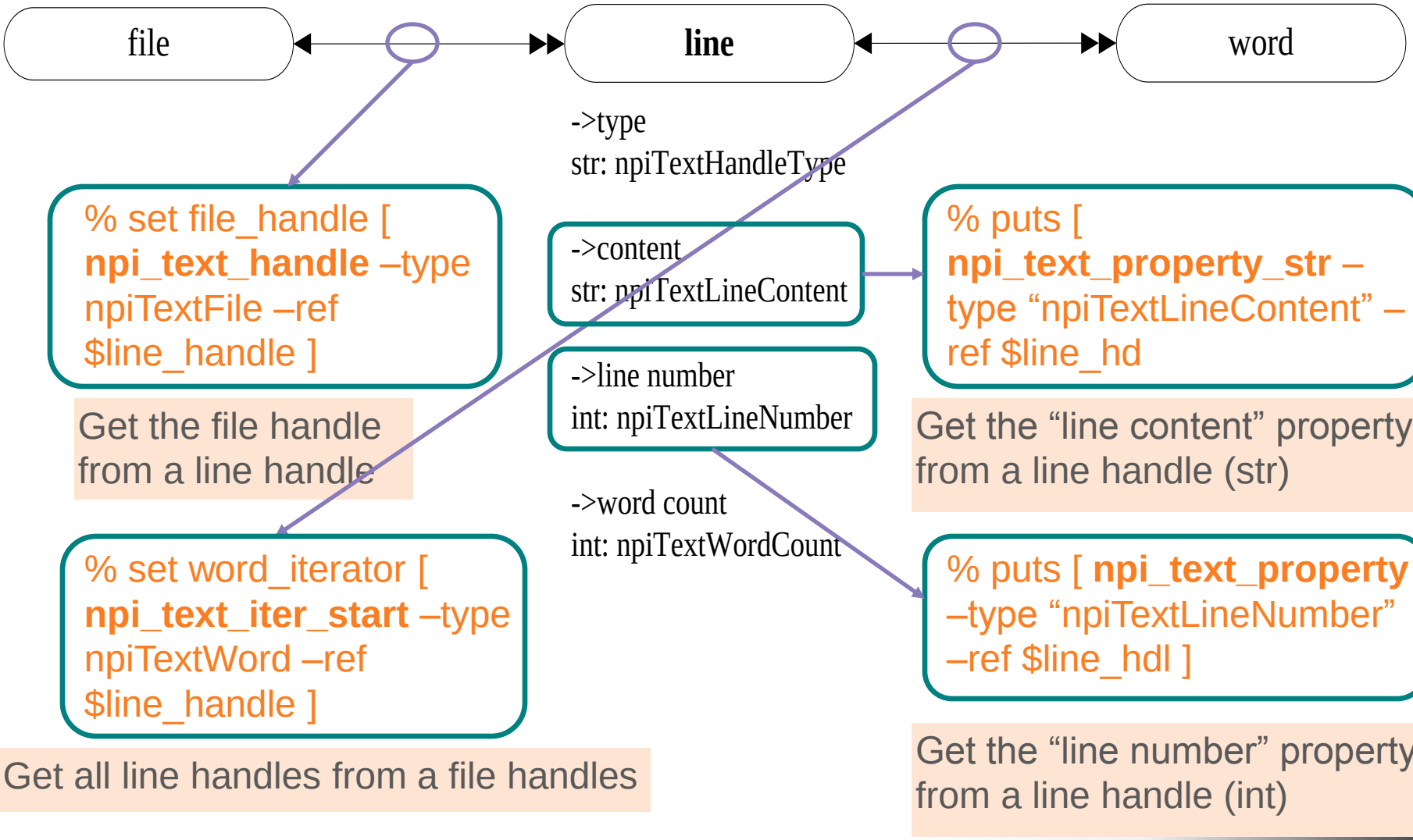
- How can I get the type string of **–type** option for **npi_text_property** and **npi_text_property_str**?
  - Refer to the object diagram according to your reference object handle, and check the description under the object diagram
    - **str**: means the property is string type.
      Use **npi_text_property_str** command
    - **Int**: means the property is integer type
      Use **npi_text_property** commands
    - **bool**: means the property is Boolean type
      Use **npi_text_property** command

Object Diagram



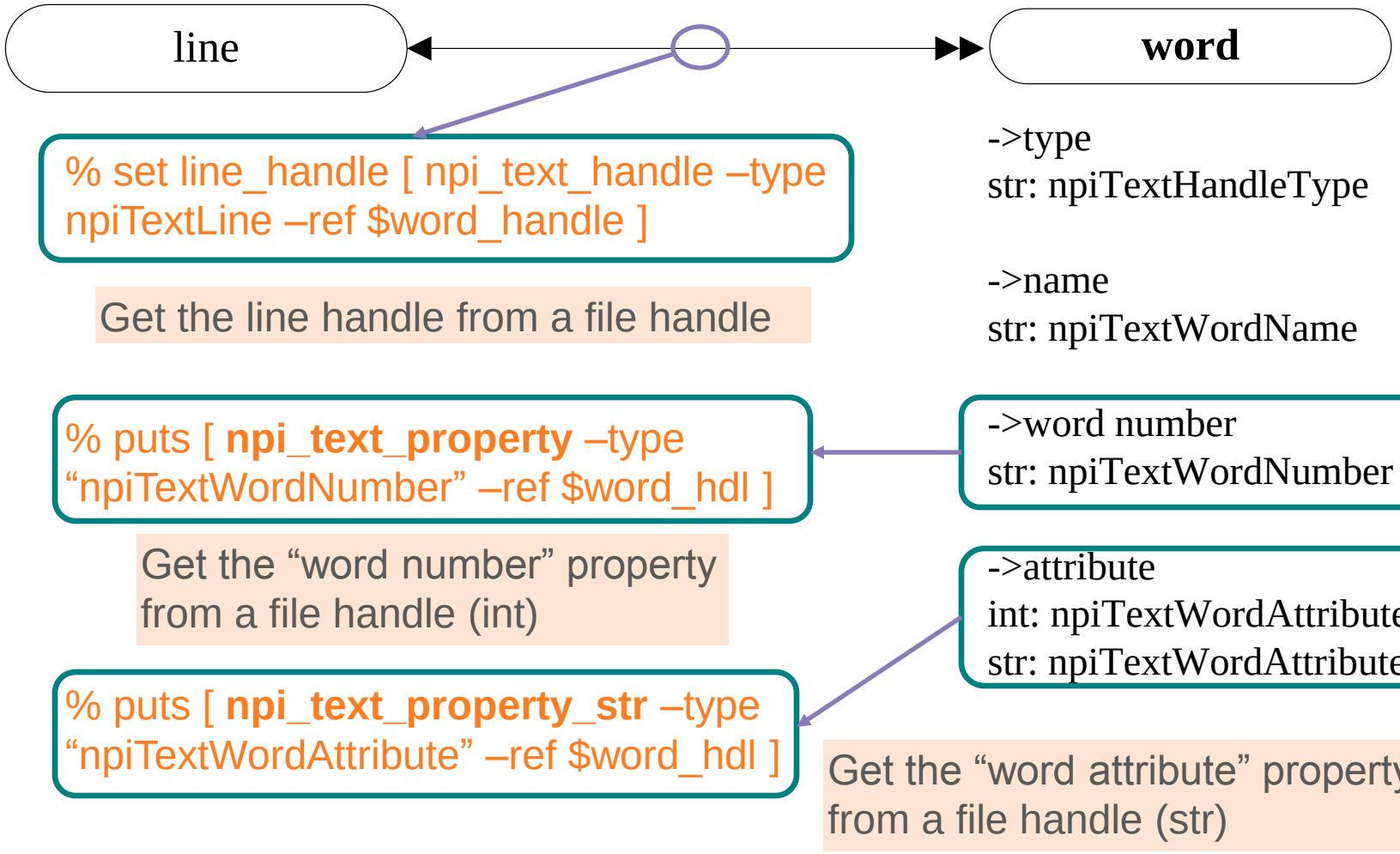
Object Diagram

Line



Object Diagram

Word



Overview

- DM Model Introduction
- NPI Commands
- NPI Object Diagrams
- Case Study

Case Study

Overview

- Requirement:
  - Modify the source code to create a input port d

- To execute the script:
  - Verdi example.v –play insert_word_before.tcl &
  - Refer to following slide for the content of **insert_word_before.tcl**

```
module test (a, b, c);
  input a, b;
  output c;
  and and_gate (c, a, b);
endmodule
```

```
module test (a, b, c, d);
  input a, b, d;
  output c;
  and and_gate (c, a, b);
endmodule
```

Case Study

Script (1/2)

```
set file_hdl [ npi_text_file_by_name –name “example.v” ]
```

Get the handle for file example.v

```
set line_iter [ npi_text_iter_start –type “npiTextLine” –ref $file_hdl ]
```

Create iterator **$line_iter** for all lines in the example.v file

```
set line_hdl [ npi_text_iter_next –iter $line_iter ]
```

Scan the *$line_iter* iterator to get the first line

```
set word_iter [ npi_text_iter_start –type “npiTextWord” –ref $line_hdl ]
```

Create iterator **$word_iter** for all words in the first line

```
while { “” != [ set word_hdl [ npi_text_iter_next –iter $word_iter ] ] } {
  if { “)” == [ npi_text_property_str –type “npiTextWordName” –ref $word_hdl ] } {
    set inserted_word_hdl [ npi_text_insert_word_before –ref $word_hdl –word “, d” ]
  }
}
```

Scan the *$word_iter* iterator, if the result is not null then check whether the word is “**)**” ; insert the word “**, d**” before the “**)**”

```
npi_text_iter_stop –iter $word_iter
```

Free the memory of iterator *$word_iter*

Case Study

Script (2/2)

```
set line_hdl [ npi_text_iter_next –iter $line_iter ]
```

Scan the *$line_iter* iterator to get the next line

```
set word_iter [ npi_text_iter_start –type “npiTextWord” –ref $line_hdl ]
```

Create iterator **$word_iter** for all words in the second line

```
while { “” != [ set word_hdl [ npi_text_iter_next –iter $word_iter ] ] } {
  if { “;” == [ npi_text_property_str –type “npiTextWordName” –ref $word_hdl ] } {
    set inserted_word_hdl [ npi_text_insert_word_before –ref $word_hdl –word “, d” ]
  }
}
```

Scan the *$word_iter* iterator, if the result is not null then check whether the word is “**;**” ; insert the word “**, d**” before the “**;**”

```
npi_text_iter_stop –iter $word_iter
npi_text_iter_stop –iter $line_iter
```

Free the memory of iterator *$line_iter* and *$word_iter*

```
set result_file [ npi_text_property_str –type “npiTextFileContent” –ref $file_hdl ]
puts –nonewline $result_file
```

Create an output file and write the modified content into the file