西安電子科技大學
XIDIAN UNIVERSITY

计算机科学与技术学院
SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY
国家示范性软件学院
NATIONAL PILOT SCHOOL OF SOFTWARE ENGINEERING

应用案例

主讲人：蔺一帅 讲师

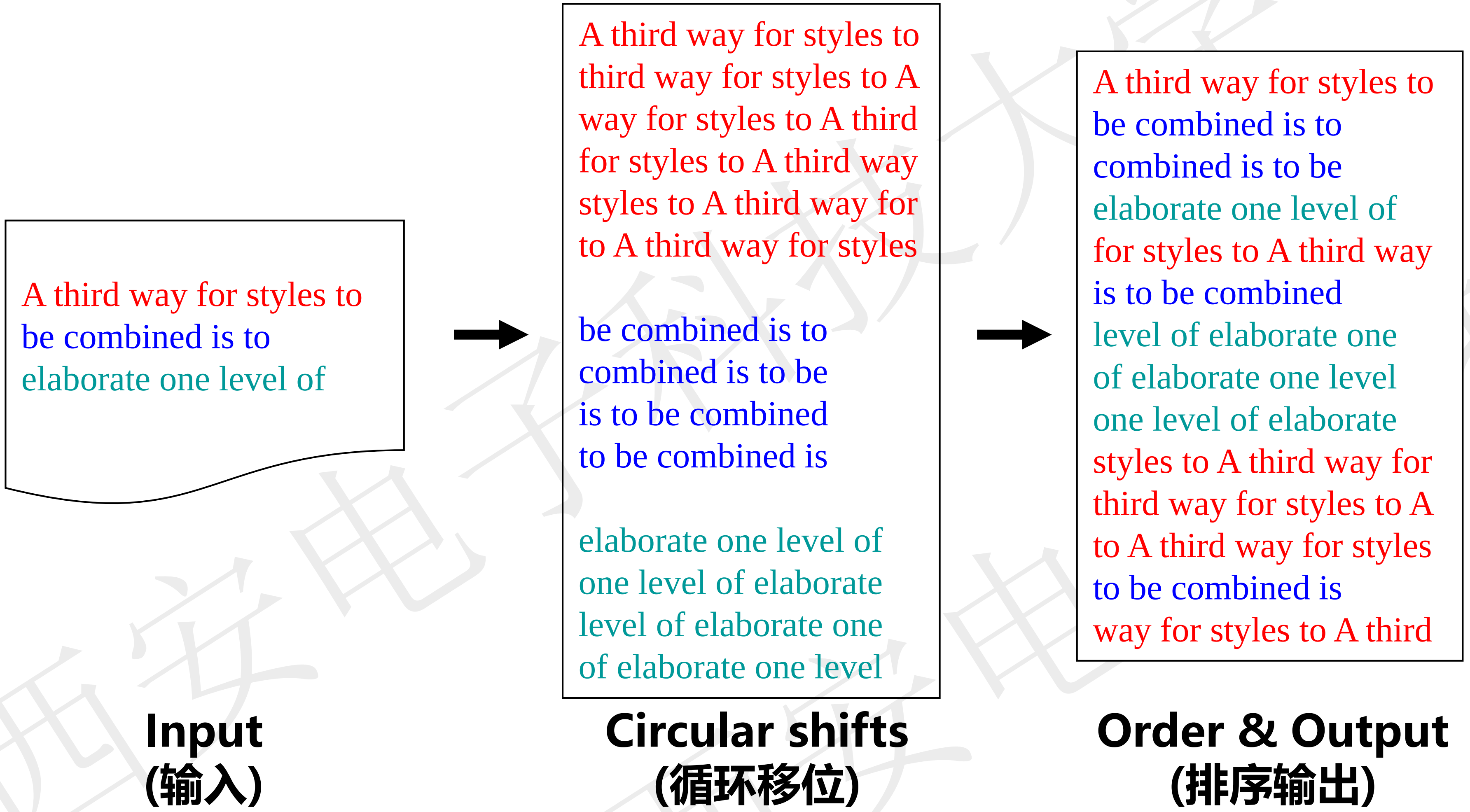
本节内容

KWIC-Functional Requirements

- In 1974, Parnas proposed the following problem,The KWIC(keyword-in-context) index system (KWIC索引系统) 软件体系结构研究的经典案例
  - ✓ accepts an ordered set of lines (接受一些行)
  - ✓ each line is an ordered set of words (每行有若干词)
  - ✓ each word is an ordered set of characters (每个词由若干字符组成)
  - ✓ any line may be "circularly shifted" by repeatedly removing the first word and appending it at the end of the line (每行都可以循环移位:重复地把第一个词删除，然后接到行末)
  - ✓ The KWIC index system outputs a listing of all circular shifts of all lines in alphabetical order. (KWIC把所有行的各种移位情况按照字母表顺序输出)

Mary Shaw, David Garlan. Software Architecture: Perspectives on an Emerging Discipline. The 3rd Chapter.

KWIC-Functional RequirementsA Case for KWIC



A third way for styles to
be combined is to
elaborate one level of

Input
(输入)

A third way for styles to
third way for styles to A
way for styles to A third
for styles to A third way
styles to A third way for
to A third way for styles

be combined is to
combined is to be
is to be combined
to be combined is

elaborate one level of
one level of elaborate
level of elaborate one
of elaborate one level

Circular shifts
(循环移位)

A third way for styles to
be combined is to
combined is to be
elaborate one level of
for styles to A third way
is to be combined
level of elaborate one
of elaborate one level
one level of elaborate
styles to A third way for
third way for styles to A
to A third way for styles
to be combined is
way for styles to A third

Order & Output
(排序输出)

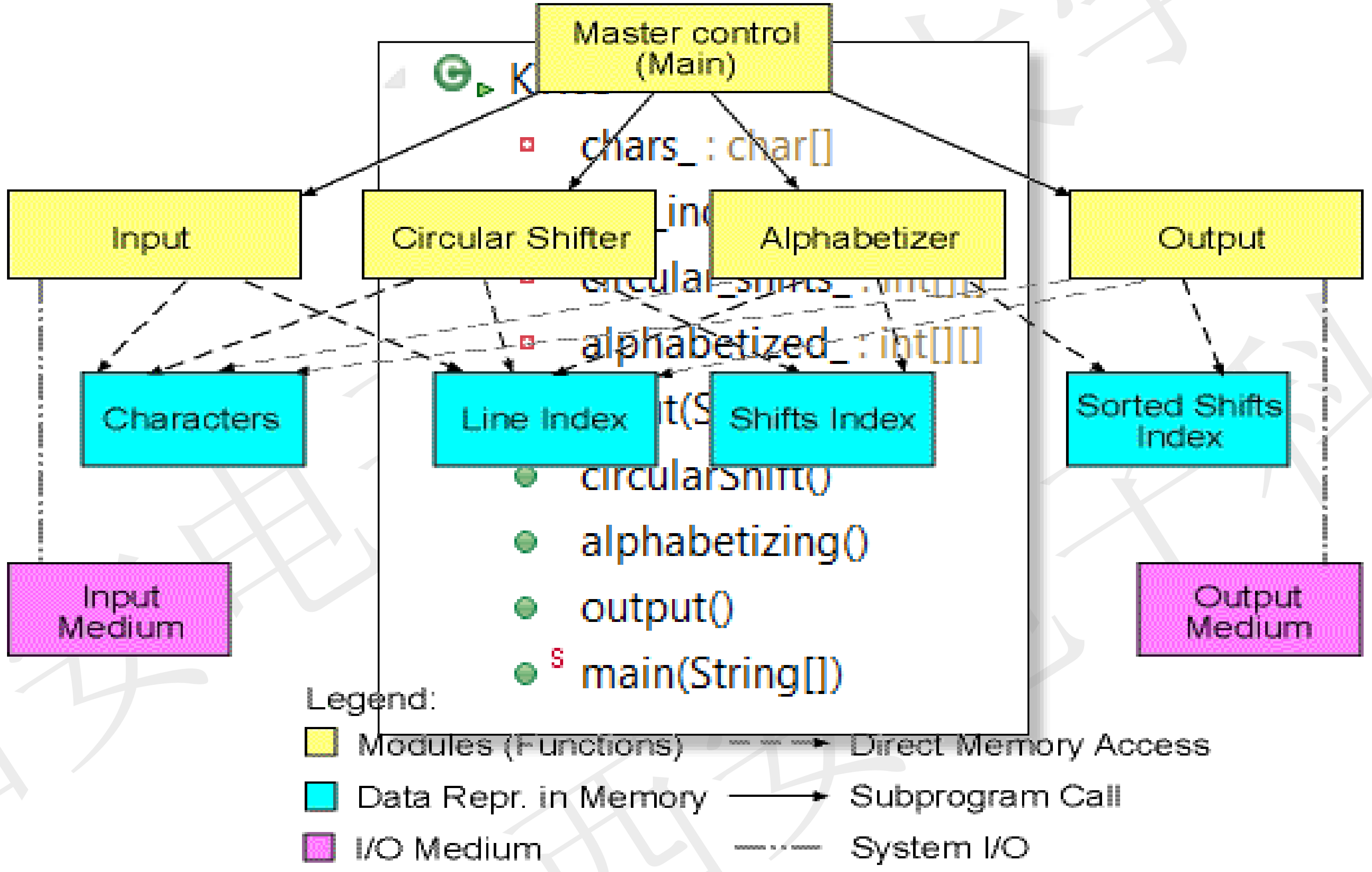
KWIC-Non-functional Requirements

- Modifiability (可修改性)
  - ✓ Change in Algorithm (算法的变化): batch vs. incremental
  - ✓ Change in Data Representation (数据表示方式的变化)
    - ✓ line storage, explicit vs. implicit shifts
  - ✓ Enhancement to system function (系统功能的可扩展性)
    - ✓ e.g., eliminate lines starting with trivial words
    - ✓ e.g., input lines from a database or from UI
    - ✓ e.g., delete, modify or add to lines from the original shifted lines
- Performance (性能): Both space and time (时空复杂性)
- Reusability (系统构件的可复用性)
  - ✓ To what extent can the components serve as reusable entities.

KWIC-Main Program/Subroutine with Shared Data

- 采用“主程序-子过程”风格、对系统进行功能分解，是最自然的想法，也是“面向过程的编程” 的主要思路。
    - Decomposes the problem according to the four basic functions performed: input, shift, alphabetize, and output. (分为四个基本功能：输入、移位、排序、输出)
    - These computational components are coordinated as subroutines by a main program that sequences through them in turn. (主程序按次序调用这四个模块)
    - Data is communicated between the components through shared storage with an unconstrained read-write protocol. (通过共享的数据存储和无约束的读-写协议在模块之间进行数据交换)

KWIC-Main Program/Subroutine with Shared Data



KWIC-Main Program/Subroutine with Shared Data

Data Presentation

- char[] chars_
  - ✓ Characters from the input file are stored in a character array named chars_. Words are delimited by a single space character.
- int[] line_index_
  - ✓ An integer array, which keeps indices of characters from the character array. Each index kept in the line index array is the index of the first character of a particular line.
- int[][] circular_shifts_
  - ✓ Two dimensional array that stores the original line number and all of its circular shifts, stored as indexes of the original character array
- int[][] alphabetized_
  - ✓ Two dimensional array that stores alphabetized circular shifts

KWIC-Main Program/Subroutine with Shared Data

Process

- The input function is called to read and parse the lines from an input file and represents it by means of the character (chars_) and line index (line_index_) array. (首先调用input函数来读取和解析输入文件并将其写在chars_和line_index_数组中)
- The main function calls the circularShift function, which produces circular shifts of each particular line and stores it in the circular shifts array (circular_shifts_). (然后调用circularShift函数，产生移位结果并写入circular_shifts_数组中)
- The main function calls the alphabetize function, which sorts circular shifts alphabetically. The result is stored in the alphabetized array (alphabetized_). (然后调用alphabetize函数对其进行排序，结果写入alphabetized_数组中)
- Finally, the main function calls the output function, which prints the sorted lines. (最后调用output函数，输出结果)

KWIC: Key Algorithm - input

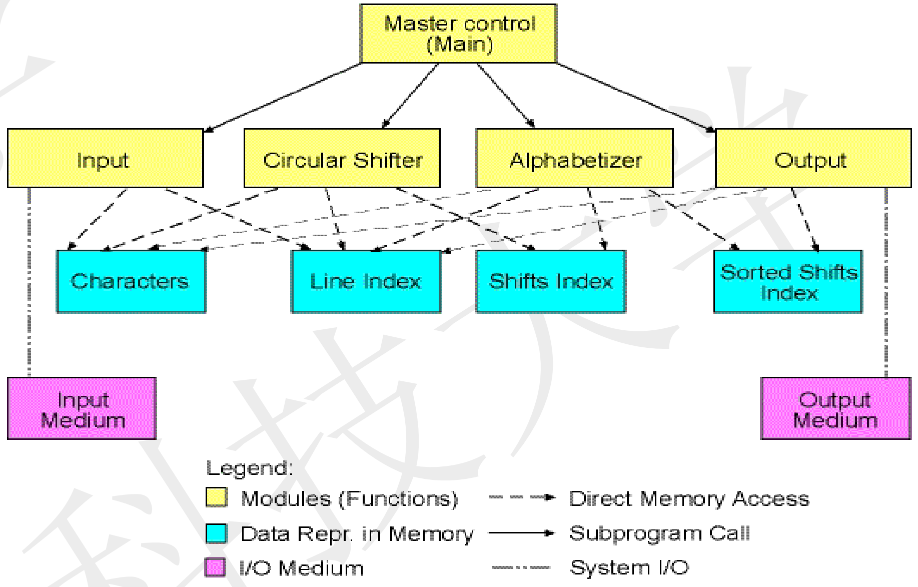
- The format of raw data is as follows.
  - ✓ Lines are separated by the line separator character(s) (on Unix '\n', on Windows '\r\n'). Each line consists of a number of words. Words are delimited by any number and combination of the space chracter (' ') and the horizontal tabulation chracter ('\t').
- The entered data is parsed in the following way.
  - ✓ All line separators are removed from the data, all horizontal tabulation word delimiters are replaced by a single space character,
  - ✓ and all multiple word delimiters are replaced by a single space character.
- Then the parsed data is represented in core as two arrays.
  - ✓ The first array is a char array (char[] chars_), which keeps all words seprated by a single space character. Since we removed line separators from the data the second integer array (int[] line_index_) keeps indexes of the chars_ array where lines start.

KWIC: Key Algorithm - circularShift

- ➢ circularShift function processes the two arrays prepared by input function.
  - ✓ It produces all circular shifts of all lines stored in core.
  - ✓ A circular shift is a line where the first word is removed from the begin of a line and appended on the end of the line.
  - ✓ To obtain all circular shifts of a line we repeat this process until we can't obtain any new lines.
- ➢ We represent a circular shift of a line as a pair of indices.
  - ✓ The first part of the pair is the index in the chars_ array, where the first word of the shifted line starts(在_array中的index).
  - ✓ To be able to determine the boundaries of the original line, the second part of the pair keeps the index of the original line in the line index array(在_line_index中的index). Thus, all circular shifts are represented as an array of such pairs, i.e., as two dimensional integer array int[][] circular_shifts_.

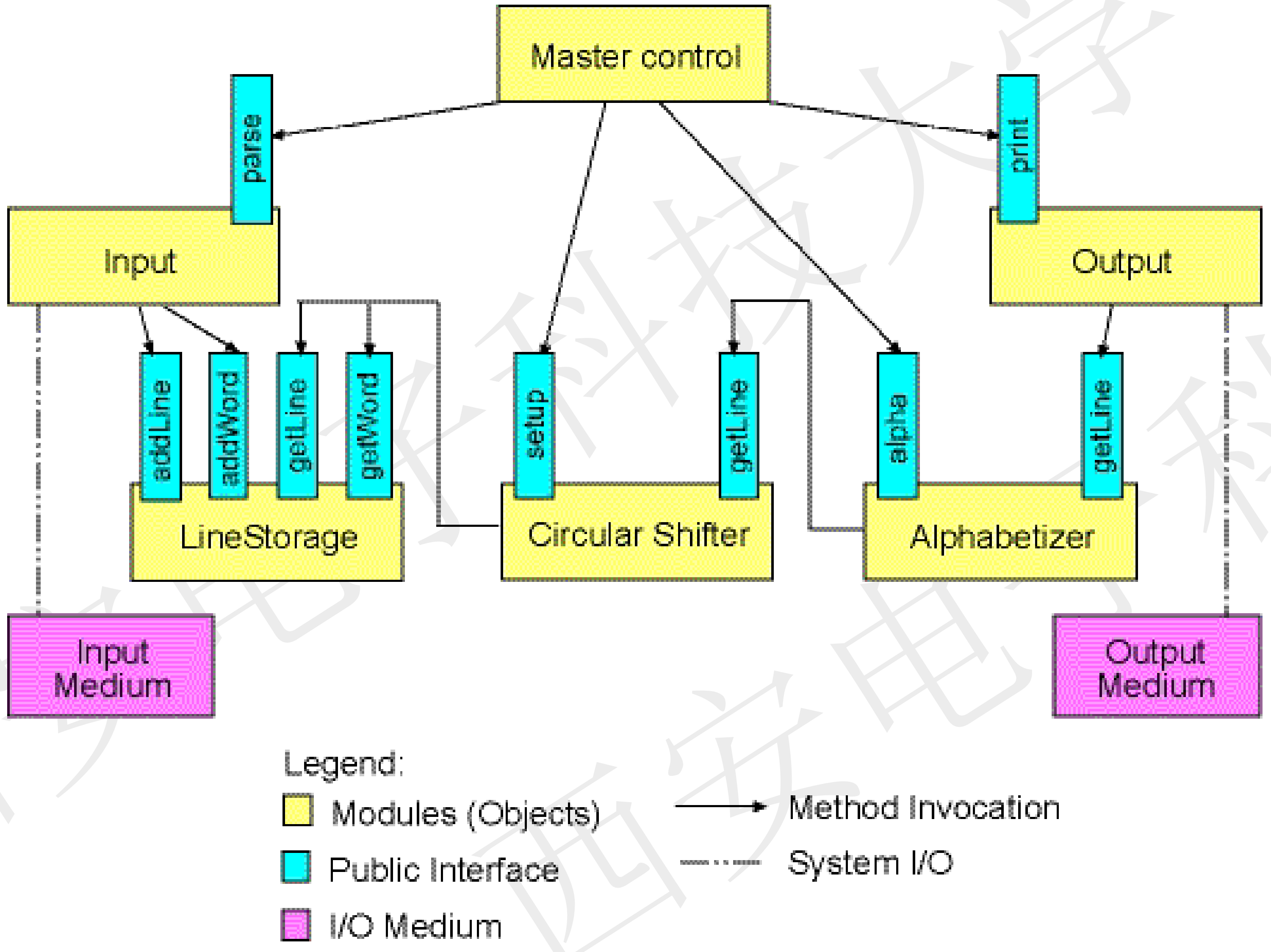
KWIC-Main Program/Subroutine with Shared Data

- **Advantages:**
  - ✓ Data can be represented efficiently, since computations can share the same storage. (模块之间的数据共享)
  - ✓ Distinct computational aspects are isolated in different modules (不同的计算功能被隔离在不同的模块中)



- **Disadvantages:**
  - ✓ A change in data storage format will affect almost all of the modules. (对数据存储格式的变化将会影响几乎所有的模块)
  - ✓ Changes in the overall processing algorithm and enhancements to system function are not easily accommodated. (对处理流程的改变与系统功能的增强也很难适应，依赖于控制模块内部的调用次序)
  - ✓ This decomposition is not particularly supportive of reuse. (这种分解也难以支持有效的复用)

KWIC-Abstract Data Types

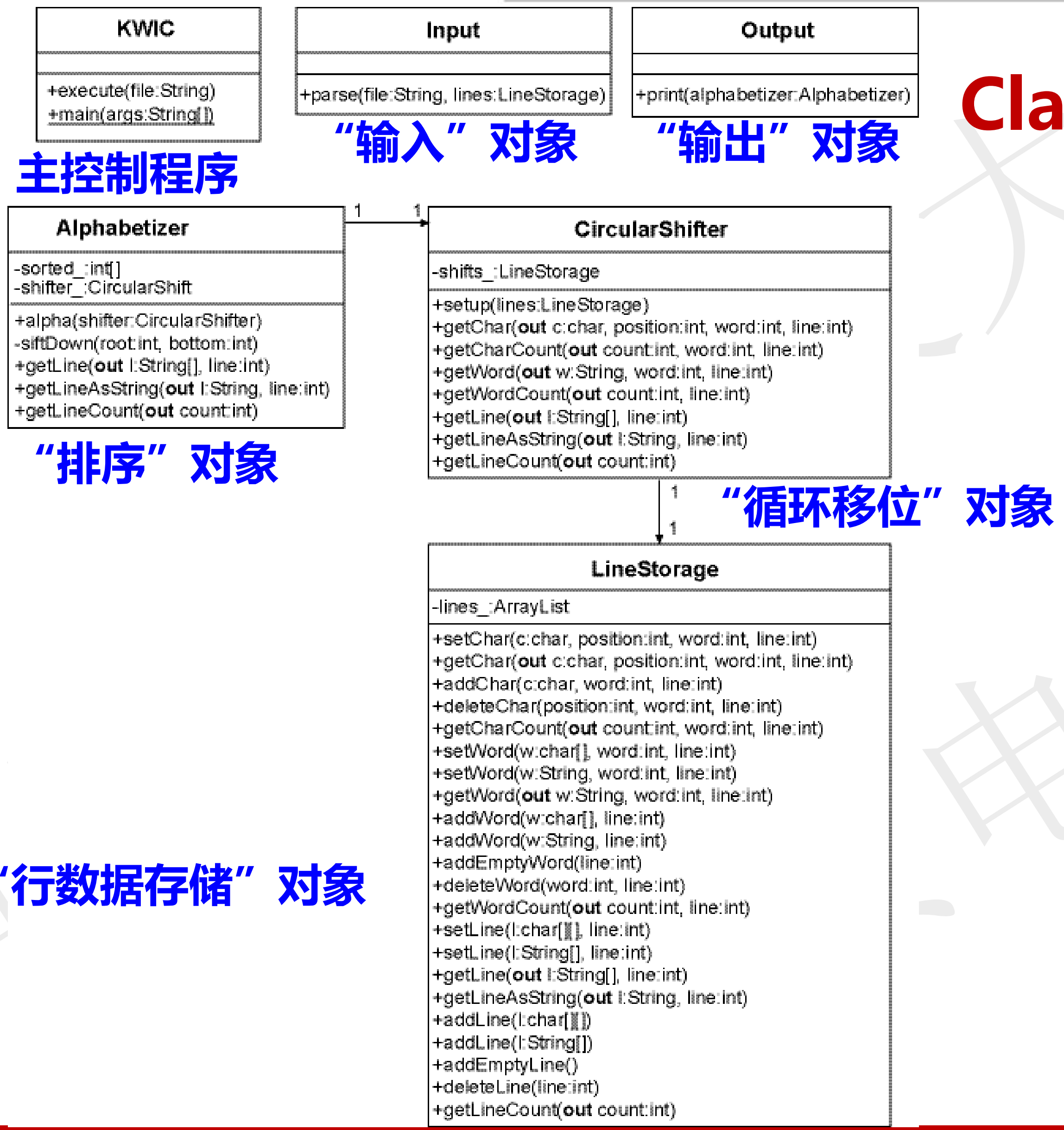


KWIC-Abstract Data Types

- 采用OO的思想，数据和作用在数据上的读写操作被封装为object，主程序调用这些object，形成控制流程；
  - Data is no longer directly shared by the computational components. (数据不再被构件直接共享，而是被封装在了Object中)
  - Each module provides an interface that permits other components to access data only by invoking procedures in that interface. (每个对象提供了一个接口，允许其他对象通过该接口调用对该对象内封装的数据的操作)

KWIC-Abstract Data Types

Class Diagram: System Statics



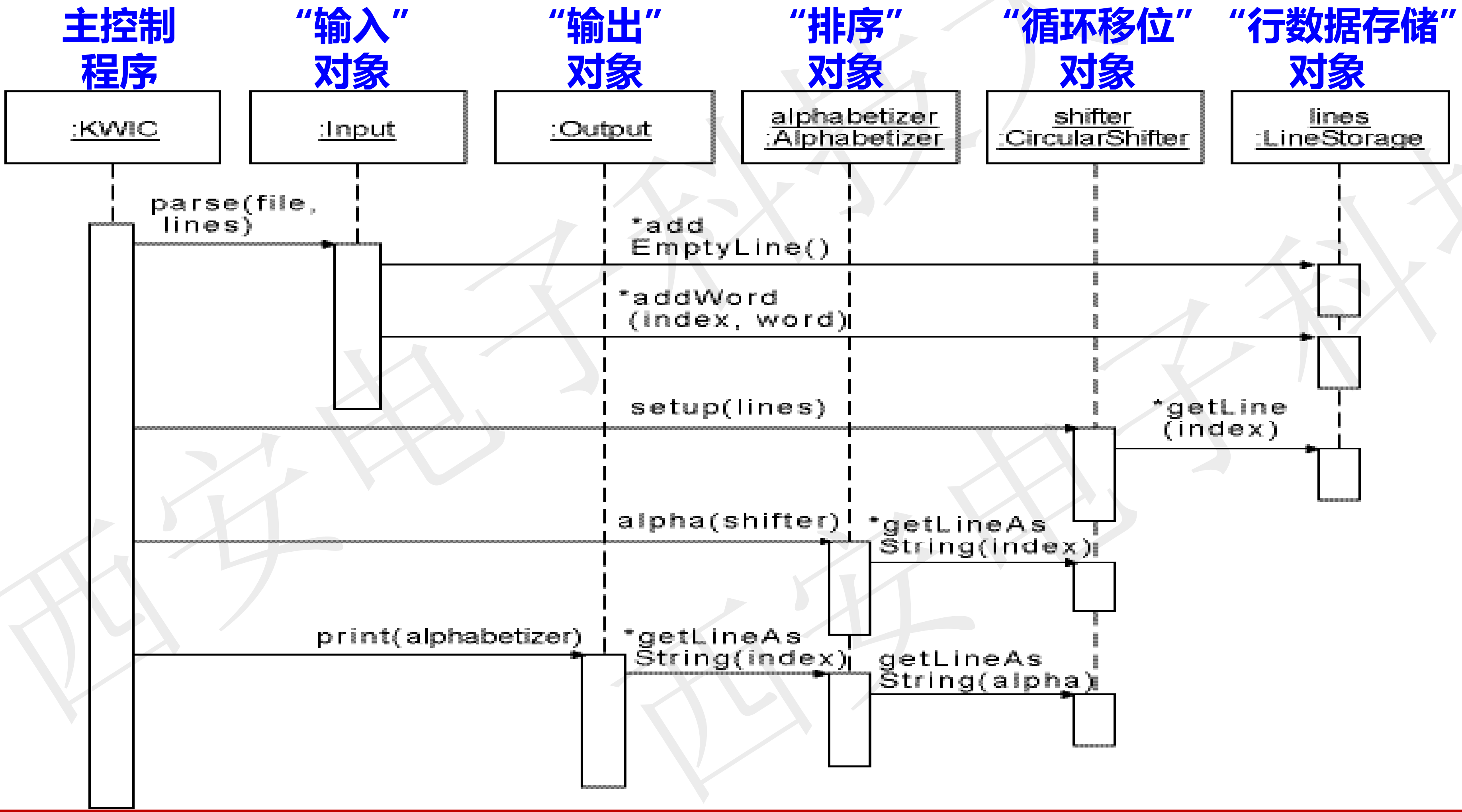
```
KWIC
+execute(file:String)
+main(args:String[])

Input
+parse(file:String, lines:LineStorage)

Output
+print(alphabetizer:Alphabetizer)

Alphabetizer
-sorted_:int[]
-shifter_:CircularShift
+alpha(shifter:CircularShifter)
-siftDown(root:int, bottom:int)
+getLine(out l:String[], line:int)
+getLineAsString(out l:String, line:int)
+getLineCount(out count:int)

CircularShifter
-shifts_:LineStorage
+setup(lines:LineStorage)
+getChar(out c:char, position:int, word:int, line:int)
+getCharCount(out count:int, word:int, line:int)
+getWord(out w:String, word:int, line:int)
+getWordCount(out count:int, line:int)
+getLine(out l:String[], line:int)
+getLineAsString(out l:String, line:int)
+getLineCount(out count:int)

LineStorage
-lines_:ArrayList
+setChar(c:char, position:int, word:int, line:int)
+getChar(out c:char, position:int, word:int, line:int)
+addChar(c:char, word:int, line:int)
+deleteChar(position:int, word:int, line:int)
+getCharCount(out count:int, word:int, line:int)
+setWord(w:char[], word:int, line:int)
+setWord(w:String, word:int, line:int)
+getWord(out w:String, word:int, line:int)
+addWord(w:char[], line:int)
+addWord(w:String, line:int)
+addEmptyWord(line:int)
+deleteWord(word:int, line:int)
+getWordCount(out count:int, line:int)
+setLine(l:char[][], line:int)
+setLine(l:String[], line:int)
+getLine(out l:String[], line:int)
+getLineAsString(out l:String, line:int)
+addLine(l:char[][])
+addLine(l:String[])
+addEmptyLine()
+deleteLine(line:int)
+getLineCount(out count:int)
```

KWIC-Abstract Data Types

- Input object (负责从输入文件中读取数据并将其存储在LineStorage对象中)
- LineStorage object (存储和处理字符、单词、行)
- CircularShifter object (负责对LineStorage对象中存储的数据进行循环移位)
- Alphabetizer object (负责对循环移位后得到的数据进行排序)
- Output object (负责打印输出排序后的数据)
- Master control object (主控制对象：负责控制其他各对象中方法的调用次序)

KWIC-Abstract Data Types

Sequence Diagram: System Dynamics



KWIC-Abstract Data Types

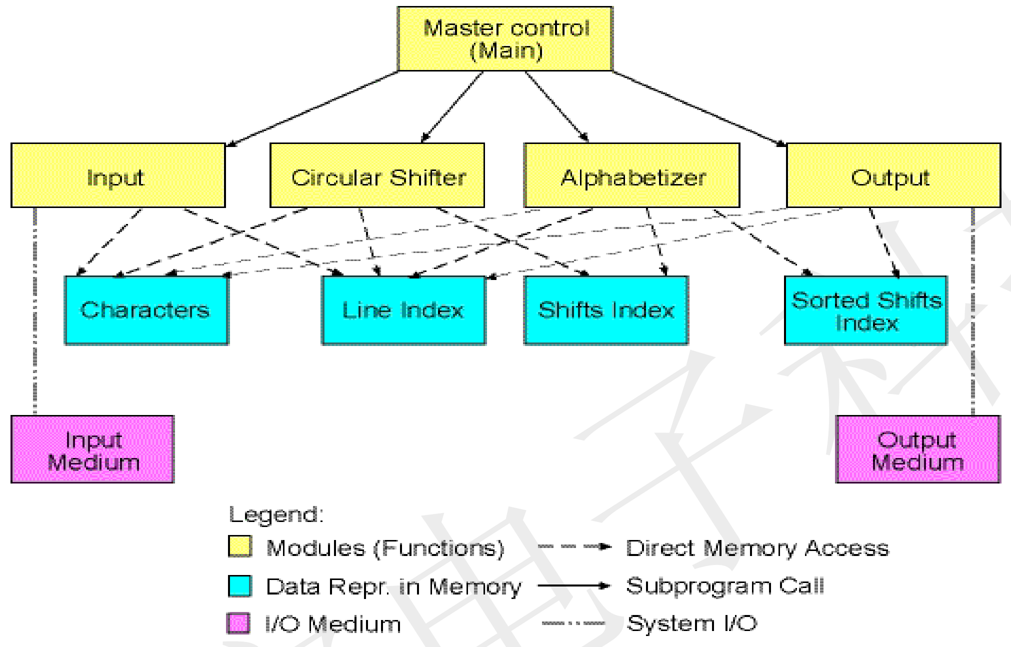
Class KWIC

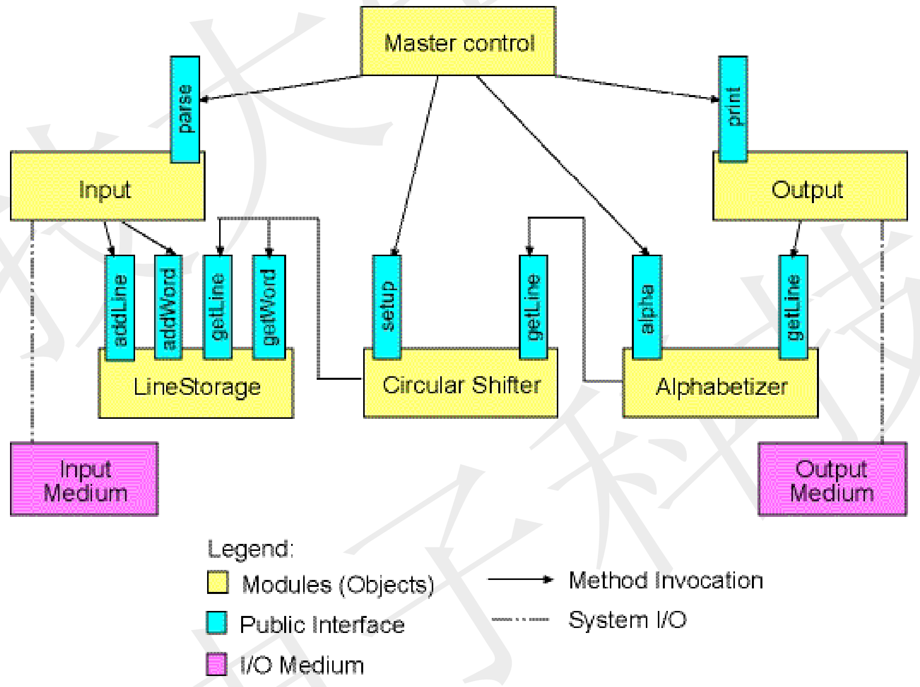
```
//主对象KWIC构造五个对象实例
LineStorage lines = new LineStorage();
Input input = new Input();
CircularShifter shifter = new CircularShifter();
Alphabetizer alphabetizer = new Alphabetizer();
Output output = new Output();
//然后分别调用这五个对象实例的某些方法
input.parse(file, lines);
shifter.setup(lines);
alphabetizer.alpha(shifter);
output.print(alphabetizer);
```

KWIC-Abstract Data Types

- **Advantages:**
  - ✓ Both algorithms and data representations can be changed in individual modules without affecting others. (某一构件的算法与数据结构的修改不会影响其他构件)
  - ✓ Reuse is better supported than in the first solution because modules make fewer assumptions about the others with which they interact. (构件之间依赖性降低，提高了复用度)
- **Disadvantages:**
  - ✓ The solution is not particularly well-suited to enhancements. (不是特别适合功能的扩展)
  - ✓ For adding new functions to the system, the implementor must either modify the existing modules or add new modules that lead to performance penalties. (为了增加新功能，要么修改已有的模块，要么就加入新的模块)

KWIC-M/S vs Abstract Data Types





从架构观点看，OO优于Main Program/Subroutine

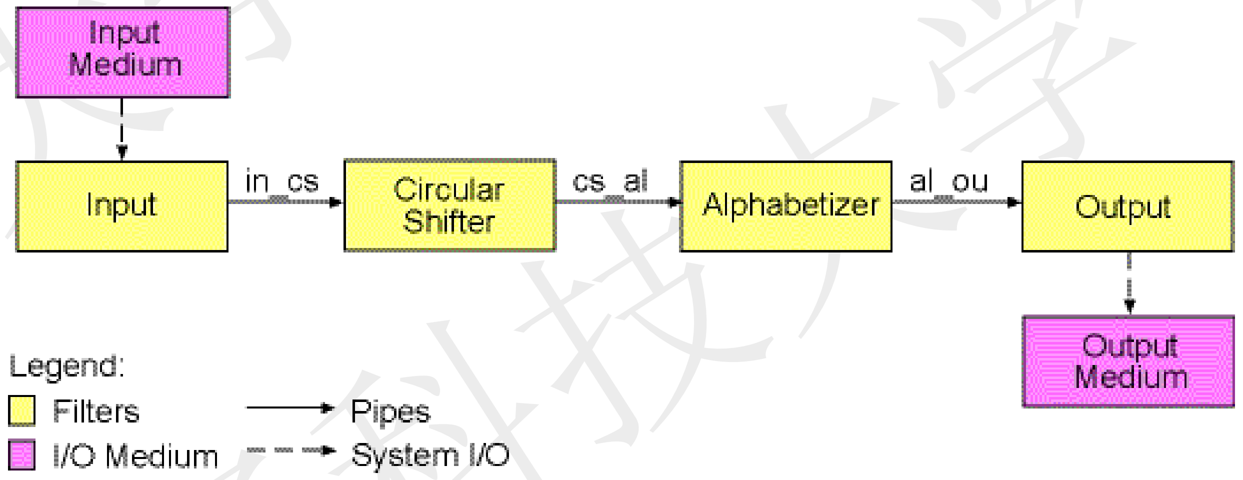
依赖减少 易于理解 易于维护 易于复用

KWIC-Pipes and Filters

- ➢ Uses a pipeline solution. (使用管道-过滤器风格)
  - ✓ There are four filters: input, shift, alphabetize, and output.（四个过滤器：输入、移位、排序、输出)
  - ✓ Each filter processes the data and sends it to the next filter. (每个过滤器处理数据，然后将结果送至下一个过滤器)
  - ✓ Control is distributed: each filter can run whenever it has data on which to compute.（控制机制是分布式的：只要有数据传入，过滤器即开始工作)
  - ✓ Data sharing between filters is strictly limited to that transmitted on pipes. (过滤器之间的数据共享被严格限制在管道传输)

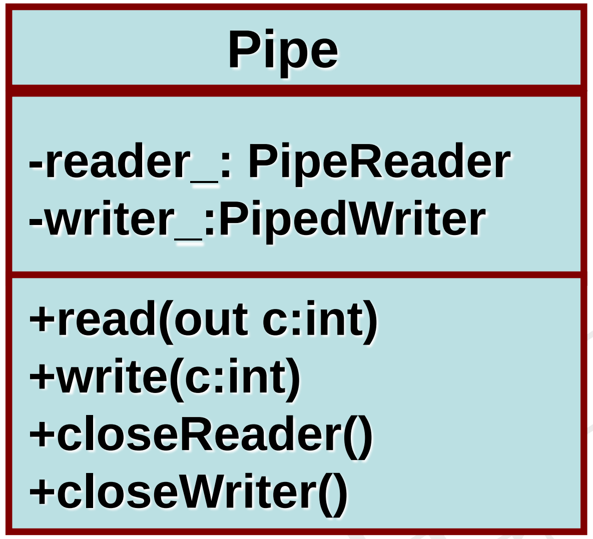
Solution : Pipes and Filters

- Filters：
  - ✓ Input filter (“输入”过滤器：从数据源读取输入文件，解析格式，将行写入输出管道)
  - ✓ CircularShifter filter (“循环移位”过滤器)
  - ✓ Alphabetizer filter (“排序”过滤器)
  - ✓ Output filter (“输出”过滤器)
- Pipe：
  - ✓ in_cs pipe
  - ✓ cs_al pipe
  - ✓ al_ou pile



KWIC-Pipes and Filters

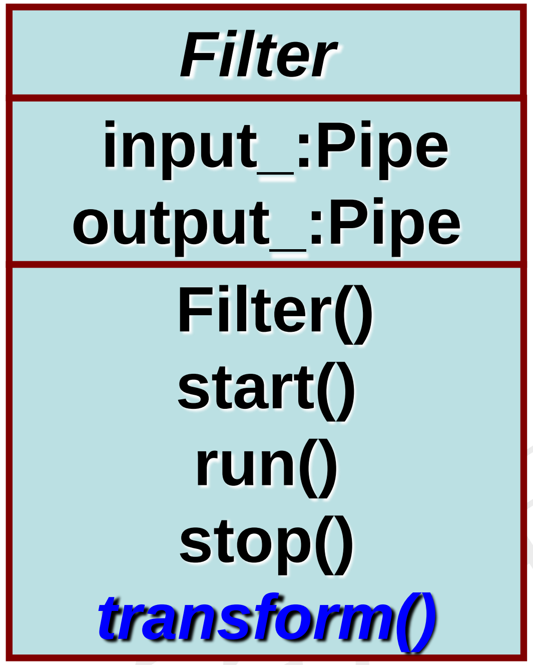
Pipe



```
import java.io.PipedReader;
import java.io.PipedWriter;
import java.io.IOException;

public class Pipe {
          private PipedReader reader_;
          private PipedWriter writer_;

          public Pipe() throws IOException {
                    writer_ = new PipedWriter();
                    reader_ = new PipedReader();   / let this piped writer to be connected to the piped reader
                    writer_.connect(reader_);
          }.    // Writes char to the piped output stream
          public void write(int c) throws IOException {
                    writer_.write(c);
          }
          //  Reads the next character of data from this piped stream
          public int read() throws IOException {
                    return reader_.read();
          }

          public void closeWriter() throws IOException {
                    writer_.flush();
                    writer_.close();
          }
          public void closeReader() throws IOException {
                    reader_.close();
          }
}
```

KWIC-Pipes and Filters

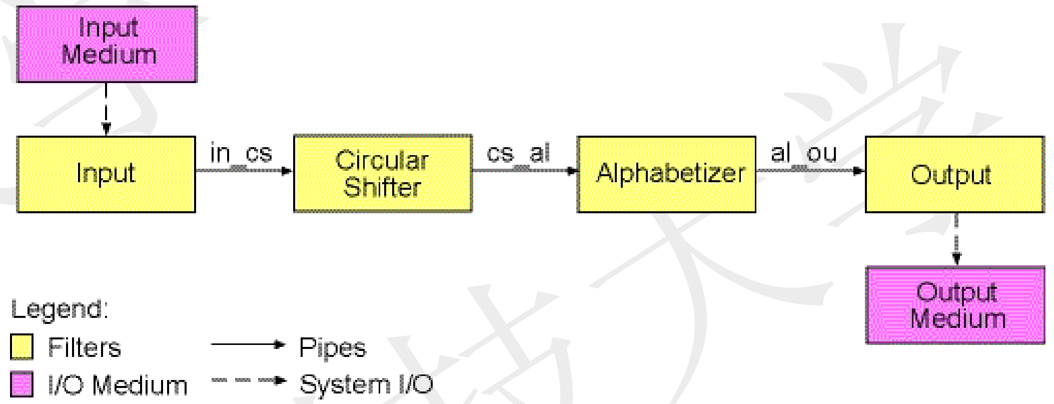
Filter



```
public abstract class Filter implements Runnable {
          protected Pipe input_;
          protected Pipe output_;
          private boolean is_started_ = false;

          public Filter(Pipe input, Pipe output) {
                    input_ = input;
                    output_ = output;
          }
          public void start() {
                    if (!is_started_) {
                              is_started_ = true;
                              Thread thread = new Thread(this);
                              thread.start();
                    }
          }
          public void stop() {
                    is_started_ = false;
          }
          public void run() {
                    transform();
          }/*
           * This method transforms the data from the input pipe and writes the
           * transformed data into output pipe.
           */

          abstract protected void transform();

}
```

KWIC-Pipes and Filters

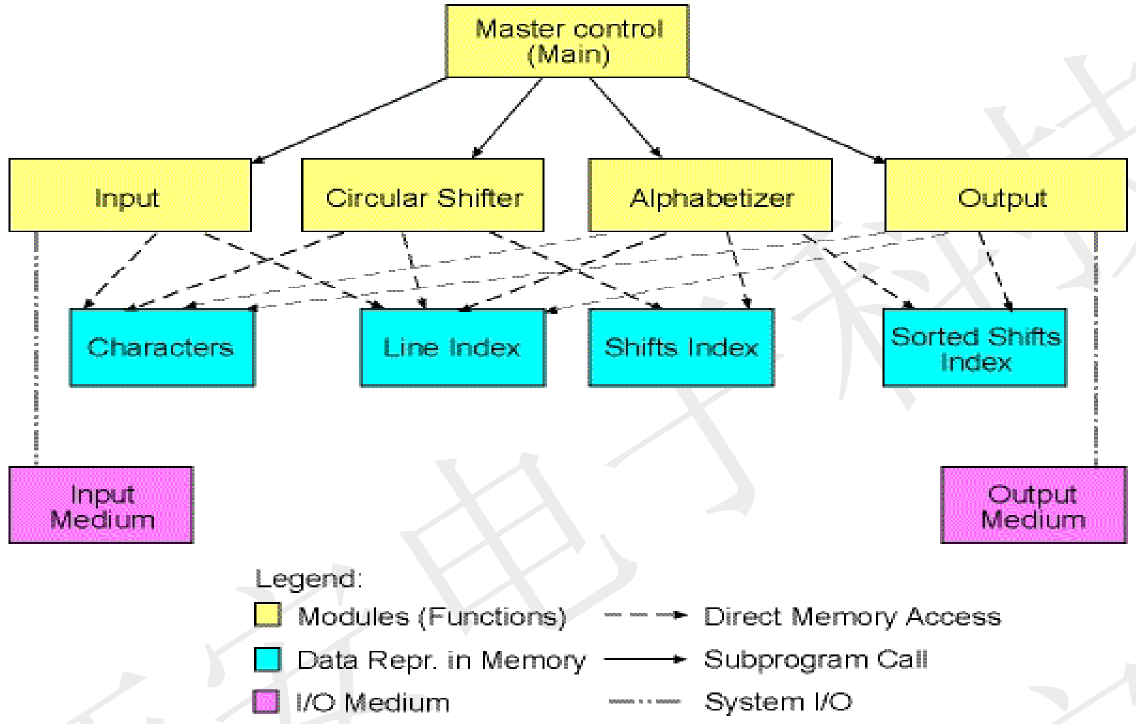
```
...

FileInputStream in = new FileInputStream(file);

Pipe in_cs = new Pipe(); // create three objects of Pipe
Pipe cs_al = new Pipe(); // and one object of type
Pipe al_ou = new Pipe(); // FileInputStream

Input input = new Input(in, in_cs);
CircularShifter shifter = new CircularShifter(in_cs, cs_al);
Alphabetizer alpha = new Alphabetizer(cs_al, al_ou);
Output output = new Output(al_ou); // output to screen
input.start();
shifter.start();
alpha.start();
output.start();

...
```

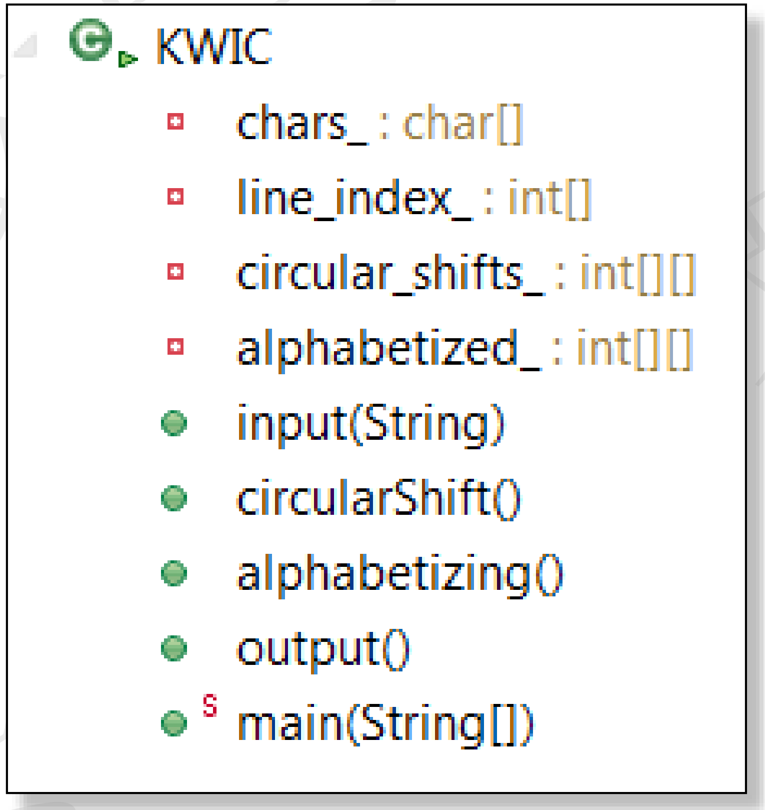


对象之间没有直接耦合

KWIC-Pipes and Filters

对比Main program/subroutine风格实现





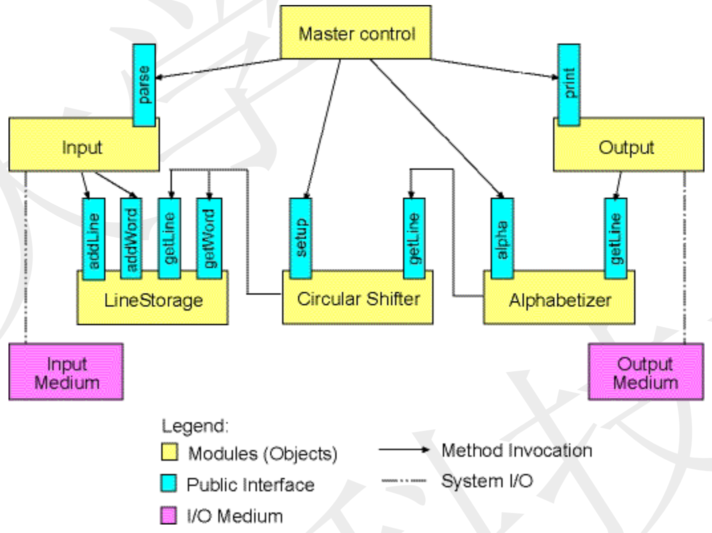
KWIC
- chars_ : char[]
- line_index_ : int[]
- circular_shifts_ : int[][]
- alphabetized_ : int[][]
- input(String)
- circularShift()
- alphabetizing()
- output()
- S main(String[])

很多很复杂的数据依赖!!!

KWIC-Pipes and Filters

对比OO风格实现

```
//主对象KWIC构造五个对象实例
LineStorage lines = new LineStorage();
Input input = new Input();
CircularShifter shifter = new CircularShifter();
Alphabetizer alphabetizer = new Alphabetizer();
Output output = new Output();
//然后分别调用这五个对象实例的某些方法
input.parse(file, lines);
shifter.setup(lines);
alphabetizer.alpha(shifter);
output.print(alphabetizer);
```



input对象依赖lines对象
Shifter对象依赖lines对象
alphabetizer对象依赖Shifter对象
output对象依赖alphabetizer对象
除此之外，还有一些间接依赖关系！

Pipes and Filters的优势

- It maintains the intuitive flow of processing. (过程流非常直观)
- It supports reuse (支持复用!!)
- It supports ease of modification, since filters are logically independent of other filters. (容易修改!!)
  - ✓ Each filter can function in isolation (provided upstream filters produce data in the form it expects).（过滤器的功能相互隔离)
  - ✓ New functions are easily added to the system by inserting filters at the appropriate point in the processing sequence. (新功能容易加入)

Pipes and Filters的不足

- It is virtually impossible to modify the design to support an interactive system. (无法支持交互式系统，局限性较大!)
  - ✓ For example, in order to delete a line, there would have to be some persistent shared storage, violating a basic tenet of this approach.
- The solution is inefficient in terms of its use of space, since each filter must copy all of the data to its output ports. (空间复杂性高)

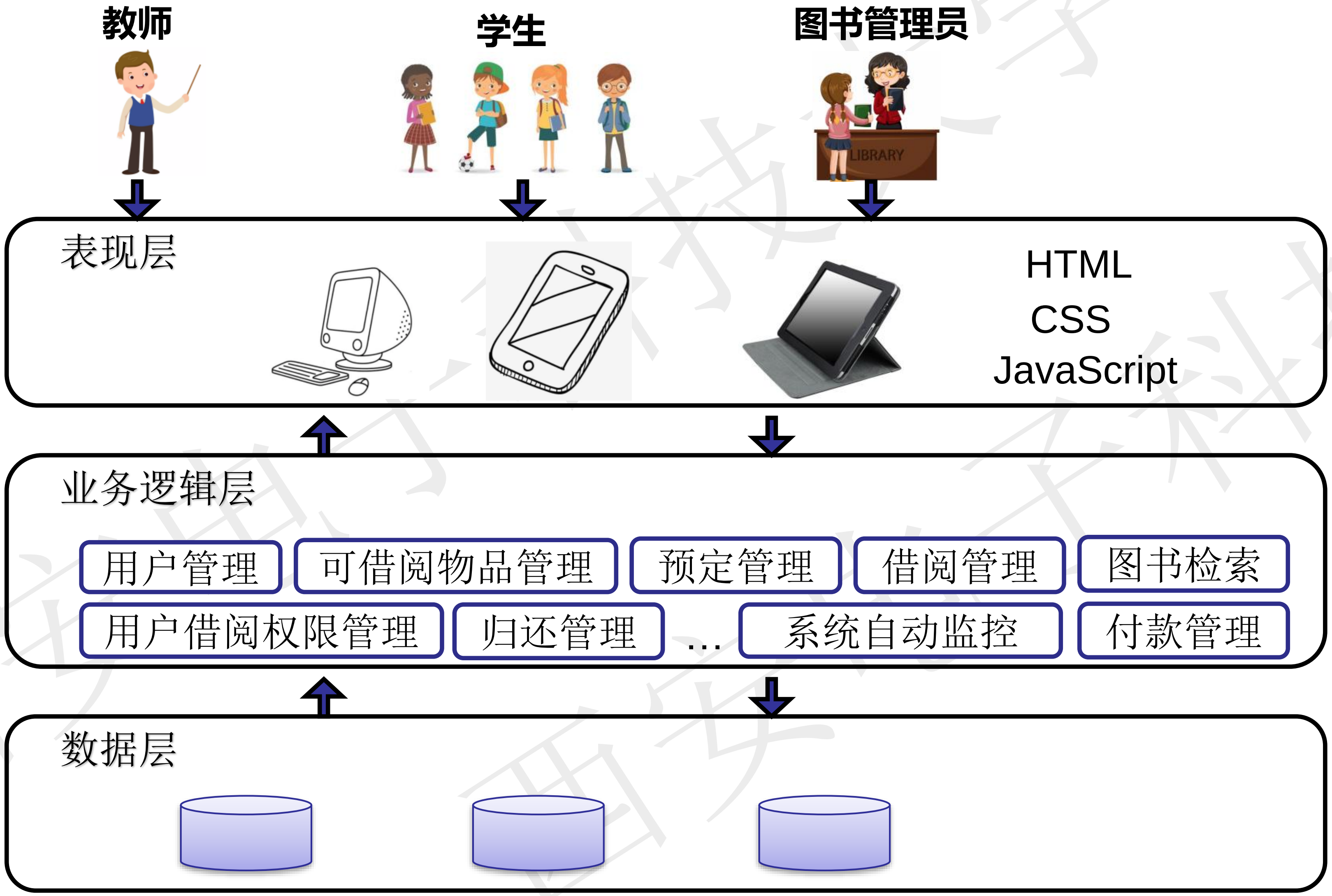
思考：有没有其他更好的架构风格呢？ Event system?

本节内容

图书借阅系统-高校图书馆系统需求

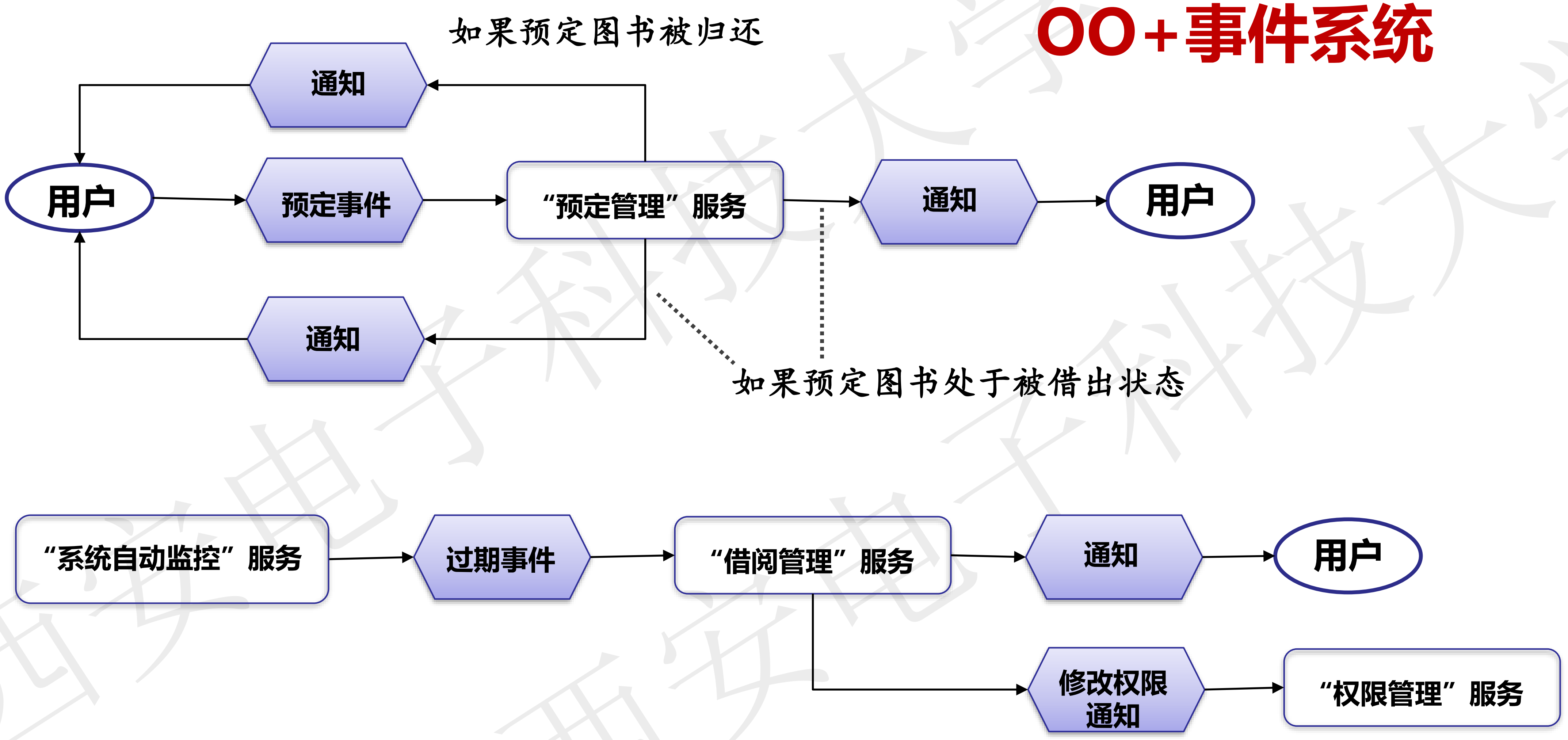
该系统向西安电子科技大学的师生提供图书借阅服务；有南校区和北校区两个图书存储地点(即2个分馆)，图书索引应能查询所有的分馆的信息；分馆之间不支持馆际借阅；学生一次可借阅16本书，1个月内归还；教师一次可借阅任何数量的书，2个月内归还；学生和教师均可以预定图书，如预定图书处于可借阅状态，预定有效期为1天；如果预定的图书正处于被借出状态，系统需要向借阅者发出一封email进行提醒；如果借阅的图书过期2周、6周、10周，则向借阅者发出email进行提醒；当过期未还时，该借阅者无法再借阅其他图书，而且当过期超过1个月后，每本书每天罚款1元；如果有未付清的罚款，借阅者也无法再借阅其他图书；除了图书之外，还提供杂志的借阅服务，借阅规则同图书相同。

图书借阅系统-架构设计


教师
学生
图书管理员
LIBRARY
表现层
HTML
CSS
JavaScript
业务逻辑层
用户管理
可借阅物品管理
预定管理
借阅管理
图书检索
用户借阅权限管理
归还管理
…
系统自动监控
付款管理
数据层

图书借阅系统-架构设计

OO+事件系统



图书借阅系统-连锁书店借阅系统需求

该系统为某连锁书店的图书借阅系统，向书店会员提供图书借阅服务；该市共有分布在不同地点的5家连锁书店，图书索引应能查询所有分店的信息；分店之间不支持跨店借阅与归还；普通会员一次可借阅10本书，3个月内归还；高级会员一次可借阅任何数量的书，6个月内归还；所有会员均可以预定图书，如预定图书处于可借阅状态，预定有效期为1周；如果预定的图书正处于被借出状态，系统会向借阅者发出一封email进行提醒；如果借阅的图书过期，系统将从过期之日起，每天向借阅者发出email进行提醒；当过期未还时，该借阅者无法再从任何分店借阅其他图书，而且当过期超过1周后，每本书每天罚款1元，从会员押金中自动扣除；如果有未付清的罚款，借阅者也无法再借阅其他图书；此外，还为高级会员提供杂志的借阅服务。

图书借阅系统-架构设计

"基于规则的系统"风格

"权限管理"服务

规则定义：

```
IF  用户类别="…" AND 项目类别="…"
THEN
    每次借阅的最大数目="…"
    最多借阅天数="…"
    超期后是否提醒="Y/N"
    超期后是否可再借阅="Y/N"
    超期是否罚款="Y/N"
    超期罚款标准="…"
    是否可预定="Y/N"
    预定请求的过期期限="…"
    预定是否付费="Y/N"
    预定的付费标准="…"
```

"用户管理"服务

规则定义：

```
IF  用户类别="…"
THEN
    需要交费成为会员="Y/N"
    费用标准="…"
    会员有效期="…"
```

图书借阅系统-借阅产品线

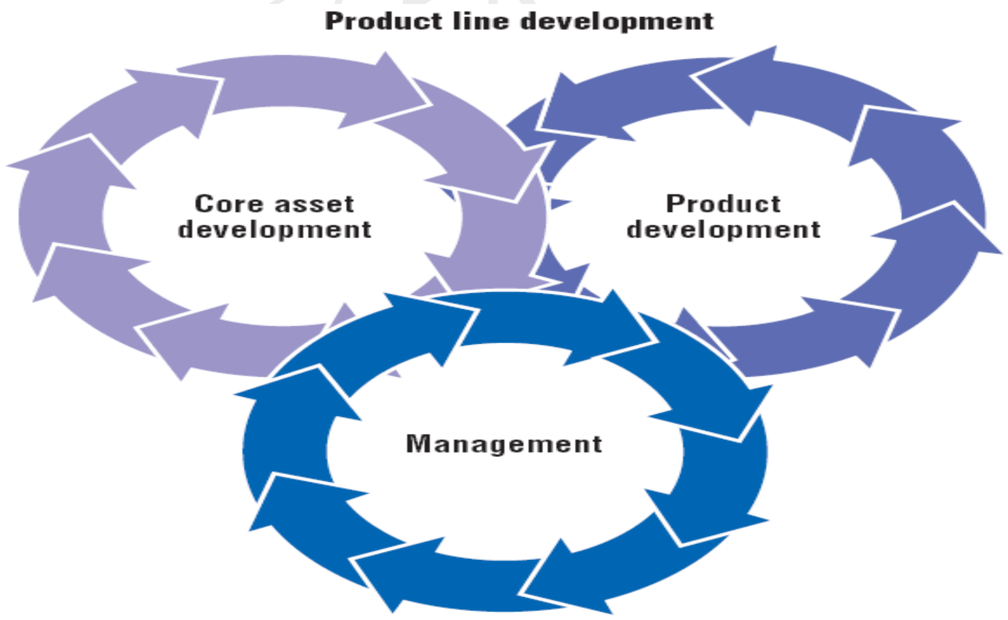
- Consists of a large number of items stored at one or more locations; (包含多个可借用的项目和多个存储位置)
- A library has customers; (有多个客户)
- Four primary actions which are performed: (主要操作)
  - 1) Removal of an item from the library by a customer (用户从库中借出某些项目)
  - 2) A customer can return an item. (用户向库还回某些项目)
  - 3) Allow customers to search for a given item by keywords; (允许用户通过关键字查询库中所拥有的可借阅项目)
  - 4) Auxiliary operations: (其他辅助操作)
    - Adding and deleting customers (注册或删除用户);
    - Adding and deleting items (增加或删除可借阅的项目);
    - Customers pay money (用户支付借阅的费用);
    - A customer may not be allowed to borrow an item (某个用户可能不允许借用某些项目);
    - A customer may place a hold on an item (用户可预定某些项目);
    - A customer can request an interlibrary loan (用户可请求一个馆际借阅);

图书借阅系统-产品线

A software product line is a set of software-intensive systems that share a common, managed feature set satisfying a particular market segment’s specific needs or mission and that are developed from a common set of core assets in a prescribed way. (软件产品线是一组软件系统，共享一组通用的特征集合，通过使用一组预先开发的/通用的核心资产来满足不同产品的研发需求)

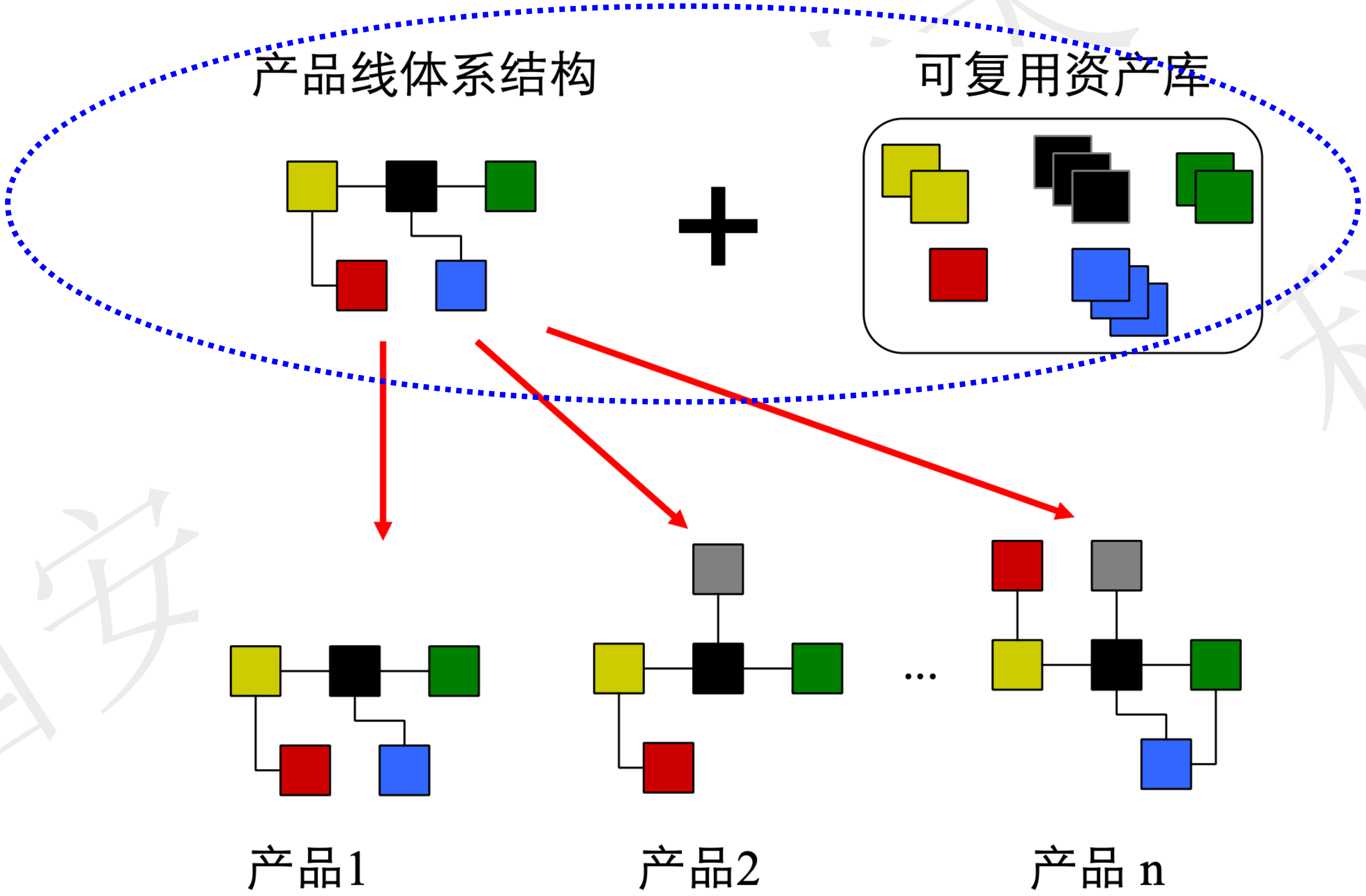
图书借阅系统-产品线

- Core asset development (核心资产开发)
- Under the aegis of technical and organizational management (核心资产管理)
- Product development using the core assets (产品开发)



图书借阅系统-产品线

Core asset development (核心资产开发)



图书借阅系统-产品线

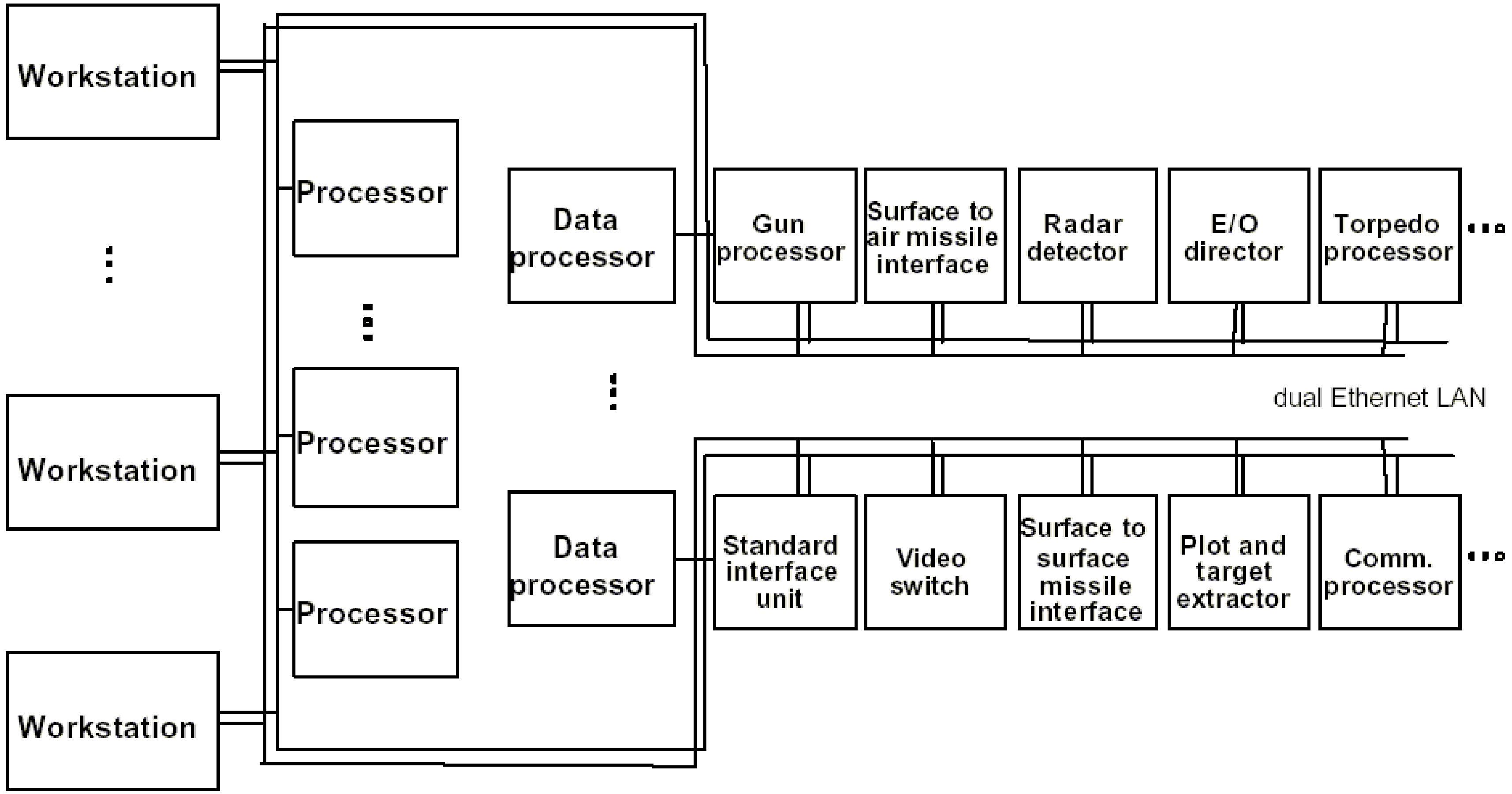
产品线的若干实践案例

- Mobile phones (移动电话)
- Command and control ship systems (舰船控制系统)
- Ground-based spacecraft systems (飞船地面控制系统)
- Avionics systems (航空系统)
- Engine control systems (引擎控制系统)
- Billing systems (帐务系统)
- Web-based retail systems (零售系统)
- Printers (打印机)
- Consumer electronic products (个人电子消费产品)

图书借阅系统-产品线

CelsiusTech的产品线

SS2000 Architecture



图书借阅系统-音像借阅系统

课后思考

有多个音像店，通过网络连接，搜索引擎需要能够搜索所有音像店的产品信息；系统支持多个店之间的互借；顾客需交纳年费才能成为会员，当会员资格即将过期时，系统向其发出缴费通知；如果会员资格过期，不允许再次借阅；音像产品按类别进行组织，但发行日期在三个月内时，将被看作“新片”；系统每天进行检索，将“新片”里不再是“新片”的video归到特定的类别里；借阅video的收费策略为：按天借阅：3元/天，但新片5元/天；按周借阅：12元/周；打包借阅：25月/3片/周；当video被归还时才开始计费，如果超期归还，按照收费标准进行额外的计费；在同一时间，一个顾客手头借阅的video最多为7片。

本案例来源于网络

西安电子科技大学
XIDIAN UNIVERSITY

计算机科学与技术学院
SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY
国家示范性软件学院
NATIONAL PILOT SCHOOL OF SOFTWARE ENGINEERING

计算机科学与技术学院微信

课程讨论群

蔺一帅 讲师 硕导

邮箱：yslin@xidian.edu.cn

主页：https://web.xidian.edu.cn/yslin/