

КОНЦЕПЦИЯ УМОЛЧАНИЯ И СИСТЕМЫ ОБУЧЕНИЯ

Стоян Бычваров

Языки общения с машиной позволяют вводить в рассмотрение те или иные объекты (например, структура данных, переменная, константа, файл). Каждому из этих объектов могут быть приписаны разные свойства, признаки, характеристики. Чем больше таких свойств, тем гибче можно управлять процессами обработки данных. С другой стороны, это приводит к усложнению процесса общения человека с машиной. При решении на машине задач из определенного класса, некоторые характеристики могут использоваться наиболее часто. Поэтому такие характеристики можно не сообщать машине, а она будет получать их в результате применения интерпретации умолчания. Если есть такая возможность, то мы говорим, что в языке имеется некоторая концепция умолчания или, что в языке применяется принцип умолчания.

Многоцелевой язык программирования естественно использовать при решении на ЭВМ широкого класса задач, ибо работа с одним многоцелевым языком имеет ряд существенных преимуществ, таких, например, как относительная простота обучения и владения (легче обучить и выучить один — пусть даже более сложный язык, чем несколько, совершено различных по строению и по используемым в них изобразительным средствам), экономия в разработке трансляторов и т. д.

К сожалению, однако, универсализация языка связана и с его усложнением. В языке неизбежно появляется большое число объектов различной структуры с различными наборами свойств. Точная спецификация всех свойств всех используемых в программе объектов становится сама по себе уже довольно трудоемкой и весьма неприятной задачей, чреватой появлением ошибок. Один из возможных путей преодоления этого недостатка является применение принципа умолчания. При этом от пользователя требуется указывать в более или менее стандартных ситуациях не все, а лишь некоторые, базовые, свойства объектов; недостающие свойства добавляются в соответствии с заранее зафиксированными правилами умолчания.

В языке PL/I [5], например, большинство выделенных объектов для нужд описания концепции умолчания обозначены идентификаторами и все идентификаторы обозначают такие объекты. Каждому объекту в результате разбора программы сопоставляется характеризующая объект конечная совокупность свойств, называемых описателями (атрибутами) объекта. В языке PL/I стандартным источником информации об объекте, обозначенном идентификатором, является описание идентификатора в

явном виде, задающее полный список его атрибутов. Принцип экономии мышления и неравномерность употребления атрибутов подсказывают, что во многих случаях восстановление полной спецификации объекта может быть произведено на основе анализа контекста, в который входит объект, и частично заданных атрибутов. Анализ контекста также результирует в построении некоторой исходной (неполной, как правило) совокупности атрибутов. Таким образом, заранее заданная концепция умолчания определяет соответствия между различными наборами атрибутов и соответствующими полными наборами.

В развитых специализированных языках также часто необходимо приписывать объектам (например, процедурам, подпрограммам) большое количество атрибутов. Поэтому как многоцелевой язык программирования, так и развитой специализированный язык можно сделать простым и удобным в использовании и обучении путем применения принципа умолчания.

Мы являемся свидетелями постоянного расширения областей применения ЭВМ. Недалек тот день, когда почти все люди будут пользователями ЭВМ в той или иной степени. Сейчас много специалистов отказываются использовать ЭВМ, либо их используют ограниченно из-за необходимости серьезного и продолжительного обучения. Поэтому проблема обучения пользователей [1, 4] будет стоять в будущем еще более остро и будет основным фактором эффективного использования ЭВМ.

Обучающая система „рассказывает“ пользователю о возможностях машины“ [2, 3], в частности, языка программирования. Исходя из вопросов и приказов человека и основываясь на заданной концепции умолчания, обучающая система знакомит пользователя только с необходимыми элементами (подмножеством) языка программирования. Применение принципа умолчания позволяет обучающей системе, по мере возрастания потребностей пользователя, расширять и углублять его знания о машине, в частности, языка программирования. Быстрое и эффективное обучение пользователей достигается путем использования подходящей концепции умолчания для конкретного пользователя или для групп пользователей. Эту подходящую концепцию умолчания описывает системный программист, учитывая особенности конкретного пользователя, в частности, класса решаемых задач. Пользователь также может изменить концепцию умолчания согласно своих потребностей.

Задание подходящей концепции умолчания можно возложить самой машине, как, например, в обучающих системах с управлением со стороны машины или с управлением со стороны машины и человека одновременно, а также в обучающе-решающих и решают-обучающих системах.

Несомненно, эффективность обучения в обучающе-решающих и решают-обучающих системах возможно повысить путем применения принципа умолчания.

Мне хотелось бы особо подчеркнуть, что существенные выгоды от использования принципа умолчания мы можем наблюдать только в сложных системах, как, например, в многоцелевых языках программирования,

развитых специализированных языках и т. д. Однако, применение принципа умолчания и в простых системах создает ряд дополнительных удобств для пользователя.

Для целей успешного использования концепции умолчания в системах программирования и системах обучения предлагаются подходящие средства описания и реализации концепции умолчания [6 — 10].

Считаем, что концепция умолчания определена путем задания функции расширения множеств $Y_i = D(X_i)$ ($i = 1, 2, \dots, d$) такой, что $X_i \subset Y_i$, где X_i — множество непосредственно указанных признаков, свойств, характеристик, возможностей объекта программы, а множество Y_i определяется транслятором (в действительности, транслятор определяет множество $Y_i \setminus X_i$).

Для описания концепции умолчания выделены следующие основные понятия: совокупность множеств альтернативных элементов, множество дополнительных элементов, каскад простых правил умолчания и дерево каскадов.

Совокупность множеств альтернативных элементов используется для определения совокупности $\{X_i, i = 1, 2, \dots, d\}$, которая содержит всевозможные аргументы указанной функции расширения множеств.

Пусть даны конечные множества M_1, M_2, \dots, M_n ($n \geq 1$) такие, что

$$M_i \neq M_j, \text{ если } i \neq j.$$

Построим множество M такое, что

$$M = \bigcup_{i=1}^n M_i.$$

Множество A называется множеством альтернативных элементов относительно совокупности $\{M_i, i = 1, 2, \dots, n\}$, если выполнены следующие условия:

$$(1) A \subset M,$$

(2) множество $A \cap M_j$ содержит не более одного элемента для любого j ($1 \leq j \leq n$).

При описании концепции умолчания нужно определить совокупность $\{X_i, i = 1, 2, \dots, d\}$. Эта совокупность может быть довольно большой (например, $d \geq 1000$). Но ее можно определить при помощи совокупности $\{M'_i, i = 1, 2, \dots, n_1\}$ всех множеств альтернативных элементов относительно совокупности $\{X_i, i = 1, 2, \dots, d\}$. При определенных условиях это может быть существенно более компактно. Поэтому при описании концепции умолчания задается совокупность $\{M'_i, i = 1, 2, \dots, n_1\}$. Она определяет совокупность $\{M''_i, i = 1, 2, \dots, n_2\}$ всех множеств альтернативных элементов относительно совокупности $\{M'_i, i = 1, 2, \dots, n_1\}$.

Множество из заданной совокупности множеств называется максимальным, если оно не является подмножеством некоторого другого множества из этой совокупности.

Так как функция расширения множеств $Y_i = D(x_i)$ ($i = 1, 2, \dots, d$) определяет концепцию умолчания, то можно положить по определению, что совокупность $\{Y_i, i = 1, 2, \dots, d\}$ состоит только из максимальных множеств совокупности $\{X_i, i = 1, 2, \dots, d\}$. Однако, иногда некоторые признаки, свойства, характеристики, возможности объекта программы считаются необязательными (дополнениями, вариантами) и следовательно, они не добавляются по умолчанию. Поэтому при описании концепции умолчания удобно допустить, что некоторые значения функции D являются подмножествами других ее значений. Для получения этой возможности вводится понятие множества дополнительных элементов. Совокупность множеств альтернативных элементов и множество дополнительных элементов используются для определения совокупности $\{N_i, i = 1, 2, \dots, h\}$ такой, что $\{N_i, i = 1, 2, \dots, h\} = \{Y_i, i = 1, 2, \dots, d\}$ и $N_i \neq N_j$, если $i \neq j$.

Совокупность $\{N_i, i = 1, 2, \dots, h\}$ называется совокупностью результирующих множеств.

Каскад простых правил умолчания используется для определения отображения совокупности множеств $\{X_i, i = 1, 2, \dots, d\}$ на совокупность $\{N_i, i = 1, 2, \dots, h\}$ (т. е. на совокупность его результирующих множеств).

Рассмотрим последовательность функций расширения множеств f_1, f_2, \dots, f_p ($p \geq 1$) таких, что

$$\begin{aligned} f_1(Z_1) &= Z_2, \\ f_2(Z_2) &= Z_3, \\ &\dots \\ f_i(Z_i) &= Z_{i+1}, \\ &\dots \\ f_p(Z_p) &= Z_{p+1}. \end{aligned}$$

Определим функцию f_i ($1 \leq i \leq p$) через пару множеств (L_i, R_i) (где $L_i, R_i \in \{X_i, i = 1, 2, \dots, d\}$ и множество R_i состоит только из одного элемента — r_i , который не принадлежит множеству дополнительных элементов) следующим образом:

Если

(i) для любого множества M'_j ($1 \leq j \leq n_1$), содержащего элемент r_i ($1 \leq i \leq p$), имеет место равенство:

$$M'_j \cap Z_i = \emptyset,$$

(ii) имеет место включение: $L_i \subset Z_i$, то Z_{i+1} полагается равным множеству $Z_i \cup R_i$. В противном случае множество Z_{i+1} полагается равным множеству Z_i .

Пару множеств (L_i, R_i) ($1 \leq i \leq p$), определяющую функцию f_i , назовем простым правилом умолчания.

Последовательность простых правил умолчания $(L_1, R_1), (L_2, R_2), \dots, (L_p, R_p)$ ($p \geq 1$), определяющую функцию F такую, что

$$(a) F(Z_1) = f_p(\dots f_3(f_2(f_1(Z_1)))) = Z_{p+1},$$

(б) область определения функции является совокупностью $\{X_i, i=1, 2, \dots, d\}$ (по определению, имеет место:
 $\{X_i, i=1, 2, \dots, d\} = \{M_i'', i=1, 2, \dots, n_2\}$),
назовем каскадом простых правил умолчания относительно совокупности $\{X_i, i=1, 2, \dots, d\}$.

Конструктивное условие (i) (т. е. условие, которое задано в терминах непосредственной его проверки) можно заменить следующим неконструктивным условием:

$$Z_i \cup R_i \in \{X_i, i=1, 2, \dots, d\} \wedge r_i \in Z_r$$

Определение функции расширения множеств $Y_i = D(X_i)$ ($i=1, 2, \dots, d$) (т. е. определение концепции умолчания) при помощи только одного каскада, как правило, является практически неудобным. Поэтому нужно ввести понятие дерева каскадов, которое позволяет удобно определить концепцию умолчания при помощи нескольких иерархически упорядоченных между собой каскадов.

Функцию расширения множеств на практике удобно определять при помощи только одного каскада, если количество множеств совокупности $\{X_i, i=1, 2, \dots, d\}$ практически намного больше, чем количество всех множеств альтернативных элементов относительно этой совокупности, т. е. если все элементы множеств X_i встречаются в разнообразных комбинациях между собой. Это обычно имеет место, когда с элементами множеств X_i связаны характеристики, признаки, свойства, возможности объекта программы, которые относятся к одному и тому классу, например характеристики арифметических переменных, характеристики строковых переменных, характеристики способа распределения памяти, области действия переменных и т. д.

Деревом каскадов называется дерево, любой вершине которого поставлен в соответствии каскад простых правил умолчания.

Функция расширения множеств $Y_i = D(X_i)$, задающая концепцию умолчания, определяется при помощи соответствующего дерева каскадов следующим образом: множество X_i определяет однозначно путь в дереве, все каскады которого применяются при определении $D(X_i)$. Точнее, с каждым каскадом связана совокупность ключевых множеств. Находим самый левый путь в дереве такой, что любой каскад этого пути имеет ключевое множество, которое является подмножеством X_i , а также любой элемент множества X_i принадлежит некоторому каскаду искомого пути. Объединение результатов применения всех каскадов найденного пути к соответствующим подмножествам X_i определяет $D(X_i)$.

Только что предложенные средства были использованы для описания концепции стандартного умолчания языка PL/I. Опыт их использования показал, что они являются практически удобными.

Предлагается концепцию умолчания, т. е. функцию $D(X_i)$ реализовать в трансляторах при помощи равенства:

$$(1) \quad D(X_i) = (X_i \subset N_1 \rightarrow N_1, X_i \subset N_2 \rightarrow N_2, \dots, X_i \subset N_h \rightarrow N_h),$$

где $\{Y_i, i=1, 2, \dots, d\} = \{N_i, i=1, 2, \dots, h\}$ и $N_i \neq N_j$, если $i \neq j$.

Равенство (1) означает следующее: если существует минимальное j ($1 \leq j \leq h$), для которого $X_i \subset N_j$, то $D(X_i) = N_j$; в противном случае X_i не принадлежит области определения функции D . Важно отметить, что порядок (слева направо) множеств N_i ($1 \leq i \leq h$) в правой части равенства (1) существен (в условном выражении).

Для языка PL/I совокупность $\{N_i, i=1, 2, \dots, h\}$ содержит около 4000 множеств.

Предложен способ компактного описания совокупности $\{N_i, i=1, 2, \dots, h\}$ в виде списка, отражающего форму представления этой совокупности (и следовательно, равенство 1)) в памяти ЭВМ. Описание совокупности $\{N_i, i=1, 2, \dots, h\}$ для PL/I занимает всего 2,5 страницы.

Те части функции $D(X_i)$, которые нельзя выразить при помощи равенства (1), либо это является неэффективным, целесообразно реализовать при помощи простых правил умолчания.

Создан комплекс программ проверки корректности описания концепции умолчания, а также преобразования описания концепции умолчания в подходящей для реализации форме.

В заключении следует отметить, что свойства, связанные с элементом множества X_i , целесообразно рассматривать в системах обучения как элементарную порцию информации, которая сообщается пользователю.

Структура описания концепции умолчания отражает структуру подачи элементарных порций информации для ознакомления с языком на разных уровнях детализации в глубине и широте.

ЛИТЕРАТУРА

1. Довгялло, А. М., Ющенко, Е. Л.: Диалоговое программирование и подготовка пользователей ЭЦВМ. Сб. Применение ЭЦВМ для автоматизации и управления учебными заведениями, ИК АН УССР, Киев, 1972.
2. Брановицкий, В. И., Кудрявцева, С. П.: Построение диалоговых решающих-обучающих систем, управляемых вычислительной машиной и пользователем. Сб. из 1.
3. Довгялло, А. М., Рыгач, В. Д.: Обучающе-решающая система для диалогового программирования инженерных задач на языке Фортран. Сб. из. 1.
4. Ющенко, Е. Л., Бабенко, Л. П., Берестовая, С. Н., Рогач, В. Д., Машбип Е. И., Балл, Г. А.: Система обучения языку Кобол, основанная на программированном пособии и ЭВМ „Днепр-21“. Сб. из 1.
5. Универсальный язык программирования PL/I. (Пер. с англ. под ред. В. М. Курочкина), М., 1968.
6. Бычваров, С. Х.: Правило умолчания и его использование в языке программирования PL/I. Сообщение ОИЯИ, II-6429, Дубна, 1972.
7. Бычваров, С. Х.: Абстрактный язык описания правил умолчания. Сообщение ОИЯИ, II-6737, Дубна, 1972.
8. Бычваров, С. Х.: Конкретный язык описания правил умолчания. Сообщение ОИЯИ, II-6738, Дубна, 1972.
9. Бычваров, С. Х.: О реализации правил умолчания. Сообщение ОИЯИ, II-6875, Дубна, 1973.
10. Бычваров, С. Х.: Описание концепции стандартного умолчания в языке программирования PL/I. Сообщение ОИЯИ, II-7100, Дубна, 1973.

Поступила 10. X. 1974 г.

DEFAULT CONCEPT AND EDUCATIONAL SYSTEMS

S. Batchvarov

(SUMMARY)

Cases for practical application of default rules are considered. Possibility to use default concept in order to simplify education are discussed. Tools for description and implementation of default rules system are proposed with regard to its realization in educational computing systems.