

აღნიშნულ კრებულში შესულია სამეცნიერო კონფერენციის **ParametricCAD'96** მოხსენებები. კონფერენცია ჩატარდა 1006 წლის 28-29 ივნისს საქართველოს ტექნიკურ უნივერსიტეტში და მისი თემა მიეძღვნა ახალი საინფორმაციო ტექნოლოგიების განვითარებას მანქანათმშენებლობაში. სხდომებზე მოხმენილ იქნა CAD/CAM-ის ჯგუფის წევრების მიერ მომზადებული მოხსენებები. შეჯამდა განვლილი წლის განმავლობაში ჯგუფში ჩატარებული სამეცნიერო-კვლევითი სამუშაოების შედეგები და მოხდა დამუშავებული ახალი პროგრამული პაკეტების დათვალიერება-პრეზენტაცია.

კონფერენციამ ჩაიარა მაღალ მეცნიერულ დონეზე, საინტერესო კამათში, აზრთა და გამოცდილების ურთიერთ-გაზიარებით, რაც აღინიშნა მონაწილეებისა და მოწვეული სტუმრების მიერ. ფორუმმა დადებითი წვლილი შეიტანა მომავალი პარტნიორული ურთიერთობების დამყარებაში.

წინამდებარე კრებულის გამოცემა განაპირობა საინფორმაციო ტექნოლოგიებისადმი დღესდღეობით არსებულმა განსაკუთრებულმა ინტერესმა და აგრეთვე აღნიშნული დარგის სპეციალისტებისათვის, როგორც ტექნიკურ უნივერსიტეტში ასევე მის გარეთაც, ანალიტიკური ინფორმაციის მიწოდების მიზანშეწონილობამ.

დიდი მადლობა გვინდა გამოვთქვათ დახმარებისათვის და მონაწილეობისათვის სტუდენტების რექტორის ბ-ნი რ.ხუროძის, პრორექტორის ბ-ნი ვ.დიდმანიძისა და მაგისტრატურის დეპარტამენტის მესვეურების მიმართ; აგრეთვე სტუმრების პროფ. გ.ბოკუჩავას, №98 კათედრის გამგის დოც. კ.დემეტრაშვილის, სხვადასხვა კათედრების დოცენტების მიმართ.

განსაკუთრებულ მადლიერებას ვუძღვნით მონაწილეობისათვის კოლეგებს კიბერნეტიკის ინსტიტუტიდან, ექსპერტული სისტემების განყოფილების გამგის, ტ.მ.კ., ქ-ნი ნ.ტყემალაძის ხელმძღვანელობით.

CAD/CAM-ის ჯგუფის

ხელმძღვანელი ტ.მ.კ., დოცენტი

ა.შარმაზანაშვილი

Основные тенденции развития интегрированных систем CAD/CAM - обзор информации InterNET

А.Шармазанашвили

CAD/CAM это интегрированная система проектирования изделий (деталей) и производственных процессов.

Задачей системы CAD традиционно является конструирование деталей и подготовка чертежей; а задачей САМ, проектирование производственных процессов обработки деталей и подготовка управляющих программ (УП) для станков с числовым программным управлением (ЧПУ).

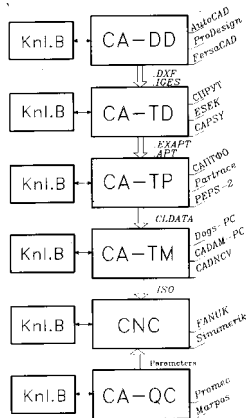
До недавнего времени эти системы развивались независимо друг от друга что усложняло их эксплуатацию из-за сложности взаимосвязки входных и выходных данных представленных в различных форматах; необходимости многократного ввода графической информации о детали и др.

Логическим исходом развития этих систем явилась их интеграция и создание сквозных каналов проектирования - начиная от конструирования деталей кончая до получения УП для станков с ЧПУ.

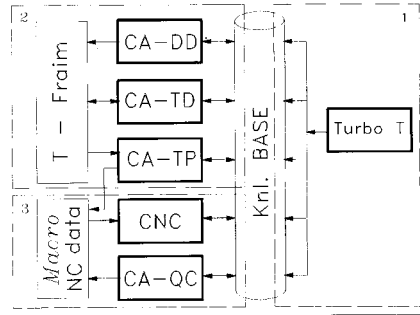
В настоящее время в мире существуют свыше 80 крупных компании разработчиков систем CAD/CAM, из которых можно выделить нескольких, которые определяют основные тенденции развития CAD/CAM. Это **Parametric Technology Corp. (PTC)**, **CAMAX**, **Autodesk**, **Intergraph**, **Computervision**, **Dassault Systems** и **Cadkey Inc.**

С точки зрения интеграции систем существуют две концепции создания сквозных каналов проектирования:

- Интеграция на основе стандартных интерфейсов, когда сквозной канал образуется в результате объединения нескольких систем. Приверженцами такой концепции являются **CAMAX**, **Autodesk**, **Intergraph**, **Cadkey**.
- Интеграция на основе общих



моделей принятия решений, когда сквозной канал образуется внутри одной системы. Примерами такой концепции являются **PTC, Computervision, Dassault Systems.**



Краткая характеристика компании; продукты; тенденции.

Parametric Technology Corp. произвела 8.200 CopyRight своего продукта, имеет 5.600 зарегистрированных пользователей.

Pro/Engineering - система разрабатываемая компанией, состоит из 28 модулей, поддерживает все рабочие станции UNIX и Windows NT, а также платформу VMS Digital Equipment Corp.

В Декабре прошлого года был анонсирован выпуск *Pro/Engineering 16.0* в которой больше внимание уделяется решению производственных задач, возникающих непосредственно в цеху - отладка УП, получение нужного прототипа.

Computervision только за 1995 год подписала контракты на сумму \$9,500,000 с фирмами AB Volvo, Jaguar, Ford Motor, HYONDAI и др. За последние месяцы получила \$21,000,000 от продажи своих Software продуктов в Аэрокосмической индустрии в Европейских странах.

CADDS - система разрабатываемая компанией стоимостью \$3,200 поддерживает рабочие станции Alpha OSF/1, 700 HP-UX, Silicon Graphics IRIX, Sun SPARC, IBM AIX/6000.

В декабре прошлого года анонсировала выпуск двух дополнительных пакетов к *CADDS 5*:

CV Design Engineering (\$25,000) для проектирования свободных геометрических форм

CV Production Modeling (\$9,900) для моделирования

конечного изделия с возможным анализом стоимости прототипа.

Dassault Systems произвела 100,000 CopyRight своего продукта CATIA/CADAM; имеет 7,000 зарегистрированных пользователей.

В декабре прошлого года анонсировала партнерство с **International TechneGroup Inc. (ITI)** для создания стандартных интерфейсов DXF, DWG, IGES, STEP.

CAMAX произвела 25,000 CopyRight своего продукта; имеет 60,000 зарегистрированных пользователей. Товарооборот за 1995 год составил \$20,000,000.

CAMAND - система разрабатываемая компанией, состоит из 30 модулей; поддерживает рабочие станции HP, IBM RISC, Silicon Graphics, Sun Microsystems и операционные среды UNIX и Windows NT.

В сентябре прошлого года анонсировала выпуск нового пакета SmartCAM 4.6 (\$12,795) с расширенными функциями оптимизации и верификации процесса перемещения инструмента.

Объявила также о партнерстве с **ITI** и использовании ее PDE/Lib (Product Data Exchange) для интеграции с различными системами CAD.

Autodesk с момента ее основания (1982) имеет рынок в 115 странах и разработала продукты на 18 языках.

AutoCAD 13.0 является 32bit программной приложением, которая разрабатывается компанией и поддерживает рабочие станции IBM, Sun, HP, Silicon Graphics и Macintosh и операционные среды DOS, Windows, UNIX.

В октябре прошлого года анонсировала выпуск пакета Mechanical Library (\$495) для системы AutoCAD в которой сгруппированы 300,000 типовых деталей от 32 ведущих Американских производителей, и выпуск параметрического редактора (\$487).

Intergraph разрабатывает систему SolidEDGE (\$5,995) для конструирования и моделирования деталей механической обработки. Поддерживает рабочую станцию TDZ300/400 3D

и операционные среды Windows'95 и Windows NT.

В ноябре прошлого года анонсировала сотрудничество с **III** для создания стандартных интерфейсов для SolidEDGE.

Cadkey Inc. разрабатывает систему CADKEY 7.0 для конструирования и моделирования деталей механической обработки. Поддерживает рабочие станции IBM, Silicon Graphics IRIS, Sun SPARC и операционные среды SGI, Sun Solaris/OS, DOS.

Развивает концепцию открытой архитектуры - COD технология (Cadkey Object Developer), когда пользователь самостоятельно может "дорабатывать" систему вплоть до изменения системы меню и графического интерфейса.

Широко развиваются также связи между отдельными системами и компаниями. Так в частности **CENTRA 2000 Inc.** разработала специальный пакет для интерактивного интерфейса между AutoCAD и Pro/Engineering; **Computervision** разработала Optega TDM связывающая CADDs с AutoCAD, Pro/Engineering и Medusa. Autodesk объявила о партнерстве с **ANSYS** в рамках проекта MAI (Mechanical Application Initiative) для встраивания в AutoCAD функции предварительной оценки конструкции деталей для механической обработки; **PTC** объявила о партнерстве с **ICAM**, ведущим производителем постпроцессорных модулей в Pro/Engineering и др.

В группе **ParametricCAD**, при лаборатории #381, оптимизации технологических процессов машиностроения (каф. #20), с 1992 года ведутся работы по разработке системы CAD/CAM по оригинальной методологии.

В основу интеграции взята концепция создания сквозного канала проектирования на базе общности моделей принятия решения. При этом для общности моделей был разработан формализм в виде Конструкторско-Технологического Модуля объединяющий конструкторские модели отдельных элементов деталей и модели производственных процессов - модели инструментальных технологических переходов, модели параметрического синтеза режимов

механической обработки, модели синтеза геометрии перемещения инструмента, модели формирования фрагментов УП и модели контроля.

В настоящее время в группе **ParametricCAD** разработана система Turbo T - программная среда для моделирования и оптимизации процессов токарной обработки, которая доведена до уровня коммерческого тиражирования.

Создан сквозной канал проектирования операционной технологии токарной обработки начиная от конструкторского чертежа детали (при заданной заготовке и схеме базирования) до получения листинга УП. В частности разработан DXF процессор (Лаборант И.Слободин); создана методология идентификации контуров детали (аспирант А.Мамамтавришвили); разработана система преобразования контуров деталей и формирования зон механической обработки (магистр А.Шония); разработана модель базы знаний экспертной системы идентификации контуров и создан соответствующий интерпретатор (магистр Д.Хабазашвили); разработана методика программирования моделей производственных процессов и созданы программные модули (магистр Л.Мегрелишвили).

დეტალის საწყისი აღწერილობის გარდაქმნა სახეთა გამოცნობის მეთოდით

ა. მამათავრიშვილი

მექანიკური დამუშავების ტექნოლოგიური პროცესების ავტომატიზებული პროექტირებისას თავდაპირველად ხორციელდება საკონსტრუქტორო ნახაზის სახით მოცემული დეტალის საწყისი აღწერილობის დამუშავება.

პროექტირების ამ ეტაპზე ხდება სხვადასხვა ამოცანების გადაწყვეტა: დეტალის კლასიფიკაცია და მიკუთვნება ამა თუ იმ კომპლექსური დეტალისათვის; საკონსტრუქტორო ნახაზიდან დეტალის ოპერაციული ნახაზის გამოყოფა და რედაქტირება, მექანიკური დამუშავების ზონების გამოყოფა და სხვა.

ტრადიციულად ამ ამოცანების შესასრულებლად გამოიყენება სპეციალური ენები. ასეთი ენების ტიპური წარმომადგენელია *ЯОД (Язык Описания Детали)*.

ანალოგიური ენები უდევთ საფუძვლად ტექნოლოგიური პროცესების ავტომატიზებული დაპროექტების (ტპ ადს) ისეთ სისტემებს, როგორებიცაა *PEPS, СПРУТ, ИСКРА* და სხვ. მსგავსი სტრუქტურები აქვთ დეტალის საკონსტრუქტორო ნახაზის აღწერის საერთაშორისო სტანდარტს *IGES* და ფირმა *AutoDESK*-ის მიერ შემუშავებულ სტანდარტს *DXF*-ს.

აღნიშნული საშუალებებით დეტალის საწყისი აღწერილობის გარდაქმნა წარმოადგენს ყველაზე შრომატევად და რთულ ამოცანას, მოითხოვს დამპროექტებლის მაღალ კვალიფიკაციას, ხოლო მიღებული გადაწყვეტილებები ხასიათდებიან დაბალი საიმედოობით. ამ ეტაპის პროცედურების მნიშვნელოვანი გამარტივება და პროექტირების, როგორც პროცესის საიმედოობის ამაღლება შესაძლებელია დეტალის საწყისი აღწერილობის გარდაქმნის განხორციელებით წინასწარ განსაზღვრული საკონსტრუქტორო პრიმიტივების საშუალებით, რომლებიც გამოხატავენ დეტალის კონტურის გეომეტრიულ თვისებებს. საკონსტრუქტორო პრიმიტივების კონკრეტული სახე დამოკიდებულია დეტალის საწყისი აღწერილობის გარდაქმნის ამოცანებზე — თუ საჭიროა დეტალის კლასიფიცირება, მაშინ საკონსტრუქტორო პრიმიტივები უნდა შეიცავდნენ კომპლექსური დეტალის კონტურს; მექანიკური დამუშავების ზონების გამოყოფისათვის კი — დეტალის კონტურის ცალკეულ ფრაგმენტებს, ანუ საკონსტრუქტორო პრიმიტივებს და ა. შ.

დეტალის კონტური შესაძლებელია წარმოდგენილ იქნას ელემენტარული ზედაპირებისაგან, რომლებიც წარმოადგენენ წრფის მო-

ნაკვეთებს, ელიფსის ან წრეწირის რკალებს. მათი ურთიერთდაკავშირების წერტილებში წარმოიქმნება ე. წ. საყრდენი წერტილები, რომლებიც სივრცეში განსაზღვრულია შესაბამისი კოორდინატებით. საყრდენ წერტილთა თანმიმდევრობა და შესაბამისი კოორდინატები ქმნიან დეტალის აღწერას კოორდინატთა სისტემაში. მაშასადამე, დეტალის კონტურის ყოველ კონკრეტულ აღწერას შეესაბამება ცხადი სახით წარმოდგენილი ვექტორი, რომლის კომპონენტებს შეადგენენ საყრდენი წერტილების კოორდინატების შესაბამისი რიცხვითი მნიშვნელობები.

ვინაიდან, საკონსტრუქტორო პრიმიტივებმა უნდა გამოხატონ დეტალის კონტურის (ან მისი ფრაგმენტის) შესაბამისი გეომეტრიული თვისებები, ზოლო ეს თვისებები, თავის მხრივ, გამოხატულია ვექტორის სახით, ამიტომ, ბუნებრივია საკონსტრუქტორო პრიმიტივებში გეომეტრიული თვისებები აღწერილი უნდა იქნეს ასევე ვექტორის სახით. თავის მხრივ, საკონსტრუქტორო პრიმიტივი გამოსახავს არა ერთი დეტალის გეომეტრიულ თვისებებს, არამედ რამოდენიმე დეტალის ან დეტალთა კლასის საერთო გეომეტრიულ თვისებებს. ამიტომ, საკონსტრუქტორო პრიმიტივებში შესაბამისი ვექტორები უნდა იქნან აღწერილი არაცხადი სახით, ფორმალური პარამეტრების საშუალებით. თავის მხრივ ფორმალურმა პარამეტრებმა უნდა უზრუნველყონ საკონსტრუქტორო პრიმიტივებში შემავალი ყველა საყრდენი წერტილის განსაზღვრა სივრცეში. მაშასადამე პრიმიტივის საყრდენი წერტილები სივრცეში განსაზღვრული უნდა იყოს არა შესაბამისი კოორდინატების რიცხვითი მნიშვნელობებით, არამედ მათი გამოსათვლელი ფორმალური პარამეტრებით. ე. ი. საკონსტრუქტორო პრიმიტივის ვექტორის კომპონენტებს უნდა შეადგენდნენ ფორმალური პარამეტრები.

ასევე ფორმალური პარამეტრებითაა საკონსტრუქტორო პრიმიტივებში აღწერილი სხვადასხვა ტექნოლოგიური პარამეტრები.

პრიმიტივების საშუალებით დეტალის კონტურის აღწერის ამოცანა შეიძლება ფორმალურად იქნას როგორც სახეთა გამოცნობის ამოცანა. სადაც დეტალის კონტური წარმოადგენს უცნობ ობიექტს ანუ გამოსაცნობ სახეს, ზოლო საკონსტრუქტორო პრიმიტივები ცნობილ სახეებს ანუ ეტალონებს. მაშასადამე, სახეთა გამოცნობის ამოცანა ჩვენ შემთხვევაში შეიძლება ჩამოყალიბდეს როგორც პროდუქცია - დეტალის კონტურის D ცხადი ვექტორული აღწერიდან D' პარამეტრიზებული აღწერის მიღება

$$I:D=>D' \quad (1)$$

ჩვენს მიერ განისაზღვრა აღნიშნული პროდუქციის რეალიზების შემდეგი ვარიანტები:

1 რ ი გ ი ს ა მ ო ც ა ნ ა . მოცემულია დეტალის კონტურის ვექტორი და მისი შესაბამისი ცალკეული ვექტორები, განსაზღვრულია კავშირები კონტურის ცალკეულ ვექტორსა და საკონსტრუქტორო პრიმიტივის შორის. ექსპერტული სისტემის მიერ განისაზღვრება საკონსტრუქტორო პრიმიტივების შესაბამისი ფორმალური პარამეტრების რიცხვითი მნიშვნელობები;

2 რ ი გ ი ს ა მ ო ც ა ნ ა . მოცემულია დეტალის კონტურის ვექტორი და მისი შესაბამისი ცალკეული ვექტორები, ექსპერტული სისტემის მიერ ხორციელდება კონტურის ცალკეულ ვექტორებსა და საკონსტრუქტორო პრიმიტივის შორის კავშირების და საკონსტრუქტორო პრიმიტივების შესაბამისი ფორმალური პარამეტრების რიცხვითი მნიშვნელობების განსაზღვრა;

3 რ ი გ ი ს ა მ ო ც ა ნ ა . მოცემულია დეტალის კონტურის ვექტორი. ექსპერტული სისტემის მიერ ხორციელდება კონტურის ცალკეული ვექტორების გამოყოფა, კავშირების დადგენა ცალკეულ ვექტორსა და საკონსტრუქტორო პრიმიტივის შორის და საკონსტრუქტორო პრიმიტივების შესაბამისი ფორმალური პარამეტრების რიცხვითი მნიშვნელობების განსაზღვრა.

პირველი რიგის ამოცანისათვის შექმნილია შემდეგი მეთოდური გეგმა:

1. საკონსტრუქტორო პრიმიტივის ცოდნის კონცეპტუალური მოდელის დამუშავება;
2. საკონსტრუქტორო პრიმიტივის ცოდნის ფიზიკური მოდელის დამუშავება;
3. ფიზიკური მოდელის ინტერპრეტატორის დამუშავება;
4. ექსპერტული სისტემის შესაბამისი პროგრამული უზრუნველყოფა.

პირველი ამოცანის შესაბამისად დამუშავდა ცოდნის ბაზის კონცეპტუალური მოდელი. ცოდნის ბაზის კონცეპტუალური მოდელი შეიცავს საკონსტრუქტორო პრიმიტივის შესახებ ცოდნის იმ ელემენტებს, რომელიც აუცილებელი და საკმარისია მოცემული პროდუქციის შესასრულებლად.

კონცეპტუალურ მოდელში ცოდნის პირველ ელემენტს წარმოადგენს საკონსტრუქტორო პრიმიტივის ნახაზი. ნახაზი გვჭირდება პრიმიტივის გრაფიკული წარმოდგენისათვის.

კონცეპტუალური მოდელის მეორე ელემენტს წარმოადგენს საკონსტრუქტორო პრიმიტივის სტრუქტურის აღწერა. სტრუქტურა საჭიროა პრიმიტივის არაცხადი ვექტორის შესაქმნელად. როგორც ავლინდნენ სტრუქტურა აღიწერება საყრდენი წერტილებით და ამ წერტილების თანამიმდევრობით.

კონცეპტუალურ მოდელში შემავალ ცოდნის მესამე ელემენტს

წარმოადგენს პრიმიტივების პარამეტრიზაციის სქემა. პარამეტრიზაციის სქემა საჭიროა პრიმიტივის საყრდენი წერტილების არაცხადი აღწერისა და კავშირის განსაზღვრისათვის საყრდენ წერტილებსა და ფორმალურ პარამეტრებს შორის. ის შეიძლება გამოისახოს მატრიცის საშუალებით, რომლის ელემენტები წარმოადგენენ ბულის ფუნქციებს f^1, f^2, \dots, f^i მატრიცის რიგი განსაზღვრულია საკონსტრუქტორო პრიმიტივების საყრდენ წერტილთა სიმრავლეზე $P'_{1-i}, P'_{2-i}, \dots, P'_{n-i} \in P'_i$ ხოლო სტრიქონი ფორმალურ პარამეტრთა სიმრავლეზე

$$\Phi_{i-1}, \Phi_{i-2}, \dots, \Phi_{i-j} \in \Phi_i$$

ცოდნის ბაზის კონცეპტუალური მოდელის შემდეგ ელემენტს წარმოადგენს დეტალის კონტურისა და საკონსტრუქტორო პრიმიტივის საყრდენ წერტილთა ექვივალენტობის სქემა. ექვივალენტობის სქემა ამყარებს შესაბამისობას ერთის მხრივ დეტალის კონტურის საყრდენ წერტილებსა $P_{1-i}, P_{2-i}, \dots, P_{k-i} \in P_i$ და საკონსტრუქტორო პრიმიტივის საყრდენ წერტილებს $P'_{1-i}, P'_{2-i}, \dots, P'_{k-i} \in P'_i$ შორის, ხოლო მეორეს მხრივ შესაბამისობას საკუთრივ საკონსტრუქტორო პრიმიტივის საყრდენ წერტილებს შორის.

შემდეგ ელემენტს წარმოადგენს საკონსტრუქტორო პრიმიტივის მოდიფიკაციები. ცოდნის ბაზაში საკონსტრუქტორო პრიმიტივი შეიძლება წარმოდგენილი იქნას ზოგადი მოდელის სახით, რომლისგანაც შესაძლებელია ცალკეული მოდიფიკაციების მიღება, რომელთაც ცოდნის ბაზაში შეესაბამებათ კერძო მოდელები. ასეთი კერძო მოდელის ამორჩევისათვის აუცილებელია ცოდნის ბაზაში ლოგიკური პირობების არსებობა. რომლებიც შეიძლება წარმოვადგინოთ Ψ პრედიკატის საშუალებით, რომელსაც აქვს რთული იერარქიული სტრუქტურა და შედგება ორი ტიპის პრედიკატისაგან: Ψ' პრედიკატი, რომელიც განსაზღვრულია პრიმიტივის საყრდენ წერტილთა რაოდენობაზე და რომლისთვისაც ადგილი აქვს ზოგადობის და არსებობის კვანტორებს და Ψ'' ტიპის პრედიკატი, რომელიც განსაზღვრულია დეტალის კონტურის კოორდინატებზე და რომლისთვისაც ადგილი აქვს მხოლოდ არსებობის კვანტორს.

ამჟამად დამუშავებულია საკონსტრუქტორო პრიმიტივის ცოდნის ბაზის კონცეპტუალური მოდელის შესაბამისი ფიზიკური მოდელი და ინტერპრეტატორი, რომელიც ასრულებს (1) პროდუქციას. პროგრამა შექმნილია C^{++} პროგრამირების ენაზე და არის ობიექტ-ორიენტირებული. მიმდინარეობს შესაბამისი ექსპერტული სისტემის მონიტორის შექმნა.

DXF процессор унифицированного интерфейса CAD/CAM

И.Слободин

Входной информацией процессора является конструкторский чертеж в формате DXF, подготовленный в одной из конструкторских САПР (например AutoCAD). Конструкторский чертеж должен удовлетворять следующим условиям :

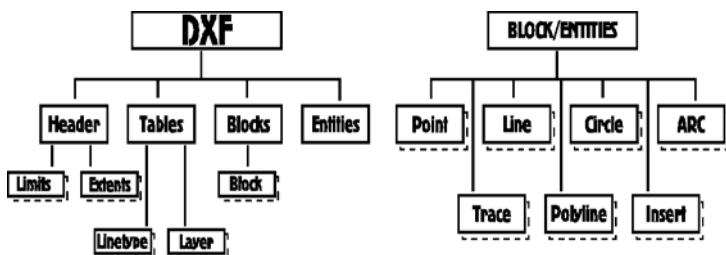
1. размеры чертежа детали должны совпадать с ее истинными размерами,
2. чертеж должен быть выполнен с достаточной точностью, так как все операции над чертежом, и соответственно, изготовление детали, производятся исходя из данных чертежа.

Процесс получения операционного чертежа можно разбить на следующие четыре этапа :

1. получение из DXF файла информации о конструкторском чертеже,
2. выделение контура детали,
3. проверка замкнутости контура; при необходимости искусственное замыкание контура,
4. образмеривание полученного контура.

Краткое описание DXF файла.

DXF файл является одним из видов описания



чертежей. Он был разработан фирмой Autodesk как интерфейсный формат для обмена чертежами между системой AutoCAD и другими САПР. В настоящее время формат DXF является международным стандартом.

DXF файл состоит из четырех отдельных секций:

1. секция заголовка (HEADER)

- 2. секция таблиц (TABLES)
- 3. секция блоков (BLOCKS)
- 4. секция объектов (ENTITIES)

Секция заголовка содержит рабочие параметры чертежа и его окружения. Наиболее полезной информацией являются координаты габаритов и границ чертежа, которые используются для определения коэффициентов масштабирования при преобразовании чертежей.

Секция таблиц содержит информацию о типах линий (непрерывная, пунктирная, штриховая и т. п.) и слоях чертежа.

Секция блоков содержит подробную информацию обо всех определенных в чертеже блоках. Блоки могут создаваться пользователем либо могут быть образованы при операциях нанесения размера, штриховки, вставки изображений и т.д.

Секция объектов содержит список всех активных графических объектов. В секции сохраняются все подробности, необходимые при воспроизведении чертежа.

Получение из DXF файла информации о конструкторском чертеже

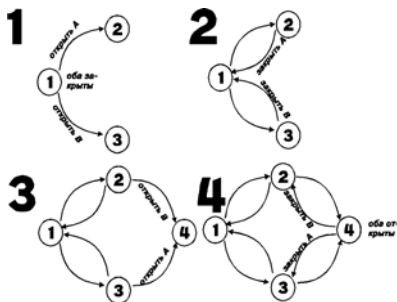
Процесс получения из DXF файла информации о конструкторском чертеже детали можно разбить на два этапа

- 1. синтаксический разбор DXF файла,
- 2. заполнение внутренних структур данных.

Для синтаксического разбора DXF файлов был применен метод конечных автоматов. Ключевыми понятиями в этом методе являются понятия состояния и события. Поставленная задача разбивается на совокупность подзадач, каждой из которых сопоставляется определенное логическое состояние. Событием является что-либо способное влиять на состояния. Программа, находящаяся в одном из логических состояний, под действием какого-либо события изменяет свое состояние, производя при действия, которые являются частью решения задачи. Конкретный вид состояний и управляющих ими событий зависит от поставленной задачи.

Поясним вышесказанное примером. Рассмотрим стол, у которого имеются два выдвижных ящика А и В. Будем открывать и закрывать ящики, причем только по-одному.

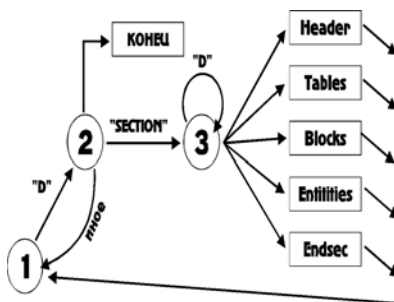
За первое состояние стола возьмем состояние в котором оба ящика закрыты. Открыв один из ящиков, мы получим еще два состояния : 2-е - открыт ящик А и 3-е - открыт ящик В. Из состояний 2 и 3, закрыв соответствующий ящик, можно перейти в состояние 1. Четвертым состоянием является состояние, когда оба ящика открыты. В него можно попасть из состояний 2 и 3 открыв второй ящик. В состоянии 4 также два возможных события - можно закрыть один из ящиков. Итого получили четыре состояния, в каждом из которых значимыми являются всего два события.



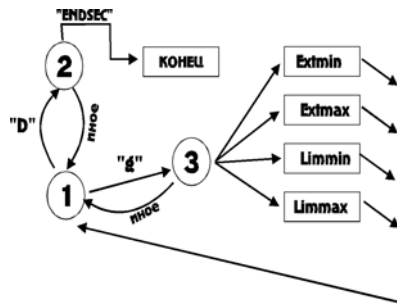
Рассмотрим, как этот метод можно применить к синтаксическому разбору DXF файлов. DXF файл представляет собой обычный ASCII файл, вся информация в котором представлена в виде текстовых строк. В файле установлено соотношение маркер - значение. Маркер является целым числом, величина которого определяет тип (целый, с плавающей точкой или строка) и основное назначение следующего за маркером значения (число, координата или имя).

Итак, под событием в данном случае понимаем значение того или иного маркера а под состоянием - возможность ввода и преобразования данных, тип которых уже определен значением маркера.

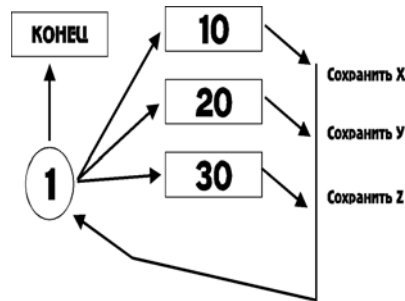
Рассмотрим конечный автомат центральной управляющей программы синтаксического разбора. Автомат обнаруживает секции и передает управление соответствующим подпрограммам разбора. При обнаружении конца файла автомат завершает работу и передает управление основной программе.



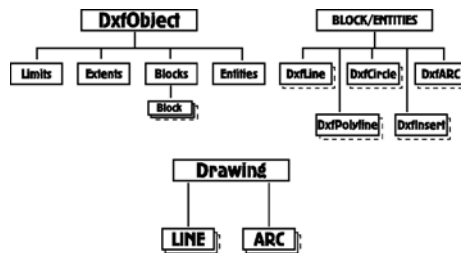
Конечный автомат разбора секции HEADER обнаруживает информацию о границах и габаритах чертежа и передает управление соответствующей подпрограмме разбора. В случае обнаружения конца секции управление передается центральной программе разбора.



Конечный автомат разбора габаритов/границ чертежа обнаруживает и сохраняет в соответствующих переменных конкретные координаты габаритов/границ чертежа.



При разработке структур данных для представления DXF файлов была применена методика объектно-ориентированного программирования. Для каждого элемента DXF файла (Block, Layer, Circle и т. г.) был разработан объект, включающий в себя все необходимые поля для представления данных соответствующего элемента. Также разработан объект, который содержит всю информацию о DXF файле в виде объектов, созданных после его разбора.



Такое представление информации наряду с важными достоинствами, такими как гибкость и удобство представления информации, обладает и существенным недостатком - усложнен доступ к отдельному элементу данных.

Один из возможных способов устранения этого недостатка заключается в следующем: необходимо разбить сложные структуры (Block, Polyline) на совокупности простейших элементов (ими являются линия и дуга) и поместить полученные данные в объект, специально созданный для этой цели. Теперь для доступа к отдельному элементу необходимо просто просмотреть их список. При выводе на экран все операции над простейшими элементами сводятся просто к изменению их размера, что резко повышает скорость программы. Данное представление также облегчает выделение контура детали.

Выделение контура детали

Выделение контура детали программным путем является неразрешимой задачей, поскольку контур ничем не выделяется среди другой графической информации в чертеже. По этой причине выделение контура предоставляется пользователю. Выделение производится в интерактивном режиме. Пользователь при помощи манипулятора "мышь" выбирает элементы контура, которые переносятся в заготовку контура детали. После завершения операции выделения, программа пытается идентифицировать полученные элементы как замкнутый контур. Если эта операция проходит успешно, управление передается подпрограмме образмеривания. В противном случае пользователь может либо продолжить выделение контура (в случае если по ошибке не все элементы были выделены), либо вызвать подпрограмму, которая осуществляет искусственное связывание контура.

Связывание контура

Связывание контура производится в интерактивном режиме. Пользователю предоставляется возможность выбрать элемент, к которому необходимо привязать несоединенный элемент.

Если незамкнутой оказалась линия, то координаты ее незамкнутого конца заменяются на координаты ближайшего к нему конца выбранного элемента. С дугой производится та же операция, причем неизменной остается величина касательной к дуге в точке, соединенной с замкнутой частью контура. При этом радиус дуги может изменяться.

Образмеривание

Образмеривание производится в интерактивном режиме. Пользователь при помощи планшета, управляемого манипулятором "мышь", выбирает систему координат, одна из осей которой - ось вращения детали.

Тем самым в контуре вводится система координат, в совокупности с которой контур и представляет собой операционный чертеж.

Спецификация

DXF процессор представляет собой приблизительно 3000 строк языка C⁺⁺. Объекты, содержащие основные алгоритмы, отделены от объектов, ответственных за оформление графического интерфейса, следовательно программа может быть легко перенесена в среду любой операционной системы.

Данная реализация программы работает в среде DOS. Исполняемый модуль занимает приблизительно 130К и требует приблизительно 300К оперативной памяти (в зависимости от размера DXF файла).

Алгоритм выделения зон черновой обработки

А.Шония

При компьютерном проектировании технологий механической обработки в начале осуществляется преобразование начального контура детали. Преобразование контура детали представляет собой процесс получения из начального контура детали промежуточных описаний, что является наиболее трудоемким этапом.

Контур детали состоит из элементарных поверхностей, которые представляют собой прямолинейные участки и дуги окружностей. В взаимосвязывающих их точках образуются опорные точки, представленные на плоскости при помощи соответствующих координат. Таким образом, контур детали соответствует вектор, компоненты которого представляют опорные точки с численными

значениями соответствующих координат.

Каждому промежуточному описанию соответствует один из векторов $\{d_1, d_2, \dots, d_n\}$, компонентами которого являются часть опорных точек D -вектора и новые опорные точки. Таким образом, задача преобразования контура детали сводится к получению из D -вектора $\{d_1, d_2, \dots, d_n\}$ векторов. Для получения этих промежуточных описаний были разработаны две стандартные процедуры:

- добавление точек в D -вектор.
- удаление точек из D -вектора.

Реально существует многообразие вариантов взаиморасположения поверхностей, следовательно, создание единой процедуры для реализации данных процедур крайне сложная и трудоемкая задача. Для решения поставленной задачи, была произведена декомпозиция, путем создания частных моделей взаиморасположения поверхностей и разработке соответствующих процедур добавления и удаления точек из вектора детали.

Модель состоит из совокупности не более трех поверхностей любого типа. В основе моделей лежит так называемая базовая поверхность. За базовую поверхность принимается та поверхность, которая содержит точку с минимальным диаметральным размером.

В результате анализа была разработана классификация частных моделей взаиморасположения элементарных поверхностей. В основе классификации лежат три признака:

1. тип базовой поверхности,
2. варианты взаиморасположения соприкасающихся поверхностей с базовой поверхностью,
3. процедурный признак

Таким образом, были получены одиннадцать частных моделей, для которых были разработаны стандартные процедуры добавления и удаления точек из вектора детали.

В случае модели 1.1. ни добавления, ни удаления новых точек не происходит.

В случае модели 1.2. добавления новой опорной точки не происходит. Удаляется



опорная точка, являющаяся общей для двух горизонтальных поверхностей.

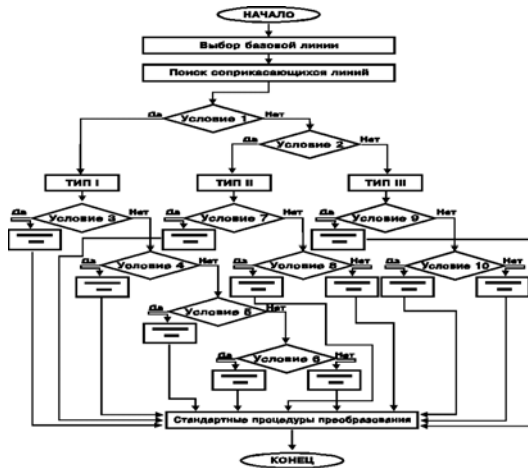
В случае модели 2.2. и 3.2. удаляется нижняя опорная точка базовой поверхности. Добавления новой опорной точки не происходит.

В случае модели 1.3. удаляются обе опорные точки базовой. Добавляется новая опорная точка на пересечении граничной прямой с горизонтальной прямой, исходящей из верхней опорной точки соприкасающейся.

В случае модели 2.3. и 3.3. удаляется нижняя опорная точка базовой поверхности. Добавляется новая опорная точка на пересечении граничной прямой и горизонтальной прямой, исходящей из верхней опорной точки базовой.

В случае модели 1.4. добавления новой опорной точки не происходит. Удаляются обе опорные точки базовой.

В случае модели 1.5. Удаляются обе опорные точки



Условие 1 - Базовая линия горизонтальна ?

Условие 2 - Базовая линия наклонна ?

Условие 3 - Соприкасающихся нет ?

Условие 4 - Одна из соприкасающихся горизонтальна ?

Условие 5 - Только одна соприкасающаяся ?

Условие 6 - У обеих соприкасающихся верхние опорные точки имеют одинаковые координаты по Y ?

Условие 7 - Верхняя опорная точка соприкасающейся, исходящей из нижней опорной точки базовой, имеет одинаковую координату по Y с верхней опорной точкой базовой ?

Условие 8 - Существует не более одной соприкасающейся, причем исходящей из верхней опорной точки базовой ?

Условие 9 - Верхняя опорная точка соприкасающейся, исходящей из нижней опорной точки базовой, имеет одинаковую координату по Y с верхней опорной точкой базовой ?

Условие 10 - Существует не более одной соприкасающейся, причем исходящей из верхней опорной точки базовой ?

базовой. Добавляется новая опорная точка на пересечении горизонтальной прямой, исходящей из верхней опорной точки короткой соприкасающейся с глинной соприкасающейся.

В случае модели 2.5. и 3.5. удаляется нижняя опорная точка базовой поверхности. Добавляется новая опорная точка на пересечении горизонтальной прямой, исходящей из верхней опорной точки базовой с глинной соприкасающейся.

Алгоритм выделения промежуточного описания по описанным процедурам имеет вид, представленный на рисунке.

Вначале из контура детали выделяется базовая поверхность, затем выделяются соприкасающиеся с ней поверхности. По типу базовой поверхности и по вариантам взаиморасположения соприкасающихся поверхностей, определяется модель и далее по описанным процедурам реализуются процедуры преобразования.

По данному алгоритму была создана программа на языке C++, версии 3.1 фирмы Борланд. Программа функционирует в среде DOS, содержит 28 000 операторов и занимает 150 Кбт дисковой памяти и предназначена для встраивания в интегрированную систему конструкторско-технологического проектирования CAD/CAM.

Интерпретатор Базы знаний в экспертной системе идентификации деталей "тел вращения".

Д.Хабазашвили

Идентификация контуров деталей является наиболее трудоемким этапом при компьютерном проектировании технологических процессов. Традиционное решение требует высокой квалификации проектировщика, а полученные результаты характеризуются низкой надежностью.

Упрощение данного этапа возможно применением

экспертных систем (ЭС), в базе знаний которой пользователь описывает типовые контура деталей на основе которых ЭС осуществляет идентификацию заданного контура.

Типовой контур (ТК) представляет собой некоторую совокупность поверхностей, в точках пересечения которых образуются опорные точки. Следовательно его геометрию можно описать вектором, элементами которого являются опорные точки и их последовательность. Эта последовательность задает направление вектора. Типовой контур является эталоном для некоторого класса контуров деталей, следовательно его геометрию надо задать неявным (формальным) видом, т.е. опорные точки должны быть определены не через явные численные значения, а описаны через некоторые формальные параметры.

Геометрические свойства типового контура однозначно определяет его схема параметризации.

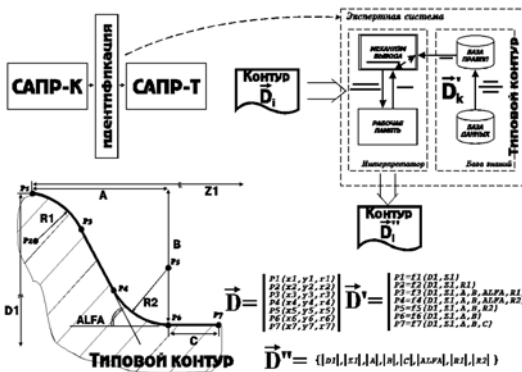
В результате применения типового контура существенно упрощается идентификация, т.к. процесс идентификации

сводится к присваиванию входного контура детали к одному из типовых контуров по геометрическим свойствам и определении численных значений соответствующих формальных параметров типового контура.

Входной контур детали может являться частной модификацией типового контура. Частная модификация - частная модель (ЧМ) типового контура имеющая общую с типовым контуром схему параметризации и различную геометрическую структуру.

Между частными модификациями и типовым контуром есть две закономерности:

Порядок расположения поверхностей. В частных модификациях некоторые поверхности типового контура выпадают из общей последовательности, которая жесткая и сохраняется всегда для всех частных модификаций данного типового контура.

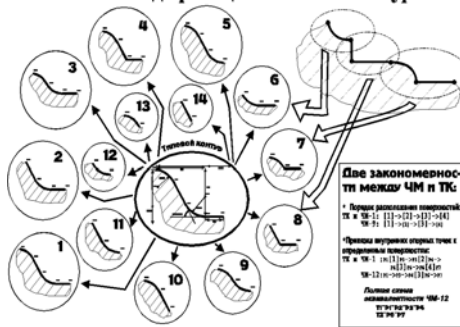


[1] → [2] → [3] → [4]
 Типовой контур (ЧМ 1)

[1] → [2] → [3] → [4]
 × ĩ - 9

Привязка внутренних опорных точек к определенным поверхностям.

В типовом контуре опорные точки, лежащие на стыках поверхностей, ограничивают определенные поверхности.



$P1 [1] P3 \rightarrow P3 [2] P4 \rightarrow P4 [3] P6 \rightarrow P6 [4] P7$ Типовой контур (ЧМ 1)

$P1 \rightarrow P3 [2] P4 \rightarrow P4 [3] P6 \rightarrow P6 [4] P7$ × ĩ - 2

Т.к. поверхность №1 отсутствует отсюда следует (эквивалентность внутренних опорных точек друг с другом):

$P1 \sim P3$

$P1 \rightarrow P3 \rightarrow P4 [3] P6 \rightarrow P7$ × ĩ - 12

Отсюда вытекает полная схема эквивалентности ЧМ-12 с ТК:

$T1 \sim P1 \sim P2 \sim P3 \sim P4$

$T2 \sim P6 \sim P7$

$Null \sim P5$

Полная схема эквивалентности частной модификации с типовым контуром позволяет однозначно сопоставить эту частную модификацию с типовым контуром

В результате возникает необходимость задания в БЗ всех частных модификаций типового контура и их полных схем эквивалентности.

Реализация указанной задачи явл. крайне трудоемкой, т.к. необходимо найти всевозможные частные модификации, а также выявить все различия между ними для однозначного описания. Это для рассматриваемого примера, типовый контур которого состоит из 4 поверхностей, потребует дополнительного описания 15 частных моделей и до 4 условий на каждую, а для случая с 7-поверхностями - 96 моделей и до 7 условий. Все это значительно увеличивает размер БЗ, требует больших объемов памяти. Процесс заполнения такой БЗ является длительным и ненадежным.

В работе был разработан метод описания элементарных

поверхностей (ЭП) типового контура в базе знаний и методика интерпретации, которое позволяет не описывать в базе знаний частные модификации типового контура, а генерировать их в результате интерпретации.

Базисным понятием в методе описания типового контура в базе знаний является элементарная поверхность, которая состоит из двух опорных точек, являющихся составной частью общего вектора контура детали. Следовательно, элементарной поверхности соответствует собственный вектор, который также является частью вектора контура детали, а опорные точки входящие в состав элементарной поверхности можно классифицировать, как начальную и конечную (CN). В результате возможна классификация элементарных поверхностей по следующим признакам:

- 1) Проекция вектора ЭП на ось X: направлена влево или вправо.
- 2) Проекция вектора ЭП на ось Y: направлена вверх или вниз.
- 3) Тип ЭП: дуга или линия.

Таким образом были сформированы 16 элементарных поверхностей.

Из полученных 16 элементарных поверхностей можно синтезировать любой типовой контур.

Для приведенной классификации схема эквивалентности для отдельных ЭП имеет вид:

$$\text{для линии: } Tz+1 \sim N \sim S \sim P\#1 \sim \dots \sim Pm$$

$$\text{для дуги: } TzR = CR = FR$$

$$\text{Null} \sim$$

$Pi+1$

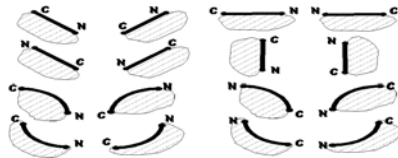
$$Tz+1 \sim N \sim S \sim P\#2 \sim \dots \sim Pm$$

В результате появляется возможность

генерации полной схемы эквивалентности частной модификации, что устраняет необходимость описания их в базе знаний.

Генерация полной схемы эквивалентности ЧМ происходит по принципу *Blackboard*. В центре которой иерархически определены гипотезы, вокруг которых сгруппированы источники знаний. Разработанная модель базы знаний имеет вид представленный на рисунке. Она представляет собой продукционную систему с использованием принципа *Blackboard*. Где гипотезы (L) - описания ЭП; а источники знаний (Q) - схемы эквивалентности ЭП.

Описание элементарной поверхности в БЗ происходит при



помощи 3 условий, которые получены из вышеописанных признаков.

$$1) (\Delta X > 0) \vee (\Delta X = 0) \vee (\Delta X < 0)$$

$$2) (\Delta Y > 0) \vee (\Delta Y = 0) \vee (\Delta Y < 0)$$

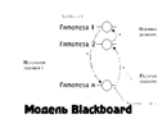
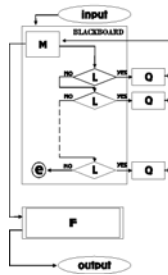
$$3) (R > 0) \vee (R = 0) \vee (R < 0)$$

где $\Delta X = C_x - N_x$, $\Delta Y = C_y - N_y$.

Дополнительно в составе контура добавляется еще одно условие, для задания порядкового номера ЭП в составе ТК. В результате для описания типового контура, состоящего из элементарных поверхностей полученных в результате классификации, понадобится система условий из n -го количества условий, где $n=4 \times j$ (j - количество элементарных поверхностей в контуре). Для рассмотренного примера понадобится $n=4 \times 64=16$ условий, а для 7-поверхностного $n=7 \times 4=28$ условий.

Пользователь вносит типовые контура в базу знаний при помощи специально разработанного языка описания конструкторского элемента. Язык описания конструкторского элемента состоит из операторов и системных переменных констант, которые позволяют быстро и компактно описать любой типовой контур, причем не изменяя старые записи в базе знаний и не вникая в все многообразие частных случаев. В языке определены два базовых типа, причем один из них составной. Используются системные и формальные переменные. В число операторов входят: оператор цикла, оператор условия, операторы описания переменных. Он позволяет использовать все математические операции и основные математические функции, такие как $abs()$, $sin()$, $cos()$, $tan()$, $asin()$, $atan()$, $acos()$, $sqrt()$...

Т.о. для полного описания типового контура в БЗ необходимо иметь описание элементарных поверхностей типового контура в предложенном виде, а также задать математические зависимости



Условия описания ЭП в Базе Знаний

- 1) $(\Delta X > 0) \vee (\Delta X = 0) \vee (\Delta X < 0)$
- 2) $(\Delta Y > 0) \vee (\Delta Y = 0) \vee (\Delta Y < 0)$
- 3) $(R > 0) \vee (R = 0) \vee (R < 0)$

и $n \times \Delta X = C_x - N_x$, $n \times \Delta Y = C_y - N_y$.

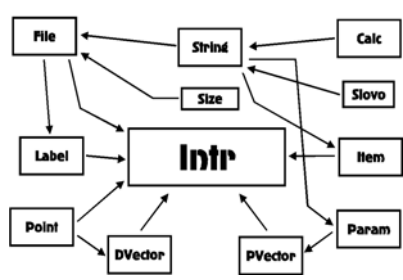
Схема эквивалентности для отдельных ЭП:

для линии: $Tx = 1 \# \# \# P1 = 1 \# \dots \# Pn$

для дуги:

```

Tx = Cr
MultiP1 = 1
Tx = 1 # P # P1 = 2 # ... # Pn
    
```



для вычисления формальных параметров.

Интерпретатор БЗ был реализован на объектно-ориентированном языке C++, в виде отдельного объекта. Всего для данной предметной области были специально разработаны 10 абстрактных типов.

Данная программа была создана в интегрированной среде программирования Borland C++ версии 3.1 фирмы Борланг. Всего было получено 24356 строк компиляции. Полученный исполнимый файл работает под операционной системой MS DOS и занимает 198,367 КБайт дисковой памяти.

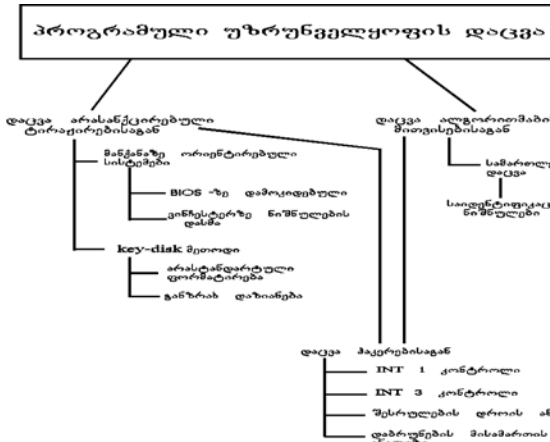
Созданная программная реализация интерпретатора выполнена в виде отдельного модуля, предназначенного для встраивания в интегрированную систему конструкторско-технологического проектирования CAD/CAM.

```
; Язык описания типового контура 3d
;
#####
; # Полуоткрытое звено #
; # Конструкторский элемент № 2 #
; # 11.03.96 #
;
#####
; геометрия контура .....
CONTOUR
0, 6, 2 ; точка P1
2, 4, 0 ; точка P3
3, 2,-2 ; точка P4
5, 0, 0 ; точка P6
8, 0, 0 ; точка P7
END
; инициализация переменных .....
VAR
koord (7)
param (Z1,D1,A,B,C,ALFA,R1,R2)
END
; автомат эквивалентности .....
LOOP
; элем.поверхность(runomezа)№1.....
IF( Cr>0 && Cy>Ny && Surface<1 )
Cr = P1r
Null # P2
N# P3# P4# P5# P6# P7
Surface = 1
END
; элем.поверхность(runomezа)№2.....
IF( Cr==0 && Cy>Ny && Surface<2 )
N# P4# P5# P6# P7
Surface = 2
END
; элем.поверхность(runomezа)№3.....
IF( Cr<0 && Cy>Ny && Surface<3 )
Cr = P4r
Null # P5
N# P6# P7
Surface = 3
END
; элем.поверхность(runomezа)№4.....
IF( Cr==0 && Cy==Ny && Surface<4 )
N# P7
Surface = 4
END
END_LOOP
; ..блокировка 15 модификации .....
IF( P12y == P1y )
; ..альтернативный выход.....
EXIT (1)
END
; ..формулы вычисления формальных параметров..
Z1 = PIX
D1 = PIY
A = ABS(6X - P1X)
B = ABS(P6Y - P1Y)
C = ABS(P7X - P6X)
ALFA =atan(ABS(P4Y-P3Y)/ABS(P4X-P3X))
R1 = ABS(PIR)
```

პროგრამული უზრუნველყოფა - დაცვა "პირატული" ტირაჟირებისაგან
კაკანეთელიძე

მოხსენების მიზანია განხილულ იქნას პროგრამული უზრუნველყოფის დაცვის ძირითადი საშუალებანი, მათი დანიშნულება და ეფექტურობის შეფასება.

ნახაზზე ნაჩვენებია პროგრამული უზრუნველყოფის დაცვის საშუალებათა კლასიფიკაცია და მეთოდები, რომლებიც გამოიყენება თითოეული ტიპის დაცვაში. როგორც



სქემიდან ჩანს, დაცვის სისტემები იყოფა განსხვავებული დანიშნულების მქონე ორ ქვეჯგუფად: I - დაცვა არასანქცირებული კოპირებისაგან და II - დაცვა ალგორითმების მითვისებისაგან.

პირველი ტიპის დაცვის სისტემებში ფართოდ გამოიყენება აპარატურაზე “მიბმული“ პროგრამული უზრუნველყოფა და ე.წ. “დისკი-გასაღებები“ (key-disks). აპარატურაზე “მიბმული“ დაცვის მეთოდებიდან ფართო გავრცელება ჰპოვა BIOS-ზე დამოკიდებულმა სისტემებმა, თანაც საკმარისად ითვლება BIOS-ის გამომშვები ფარმის და გამომშვების თარიღის კონტროლი, თუმცა ეფექტურობის ასამაღლებლად ხშირად გამოიყენებენ მთელი BIOS-ის “საკონტროლო ჯამს“. კიდევ უფრო მაღალი ეფექტურობით გამოირჩევა ხისტი დისკზე (“ვინჩესტერზე“) ნიშნულების დასმის მეთოდი. MS-DOS სისტემაზე ბაზირებულ კომპიუტერებზე ჩვეულებრივ Partition Table-ს (cyl 0, hd 0, sec 1) და პირველი ლოგიკური დისკის Boot Record-ს (cyl 0, hd 0, sec 1) შორის არის გამოუყენებელ სექტორთა დიდი რაოდენობა (სამოცამდე), სადაც შეიძლება პროგრამისათვის აუცილებელი ინფორმაციის შენახვა. აქ ჩვენ ვაწყდებით ორ პრობლემას: ზოგიერთი Boot Manager-ი იყენებს ამ მოცულობას თავისი მიზნებისათვის და თანაც ვინჩესტერის ფიზიკური ფორმატირების შემთხვევაში იკარგება შენახული ინფორმაცია და პროგრამა

მოითხოვს რეინსტალაციას.

Key Disk -ების გამოყენებისას იგულისხმება რომ არ უნდა მოხდეს ამ დისკების კოპირება. ერთ-ერთი მისაღები ვარიანტია დისკების არასტანდარტული ფორმატირება; შესაძლებელია ბილიკების ნუმერაციის არევა, “უცნაურ“ ნომრიანი სექტორების ჩამატება, სექტორების არაპროპორციული განლაგება და სხვა. თუმცა ოპერაციული სისტემის მეშვეობით ასეთი დისკების კოპირება შეუძლებელია, არსებობს სპეციალური პროგრამები, რომლებიც იძლევიან თითქმის ყველანაირი დისკების გადაწერის საშუალებას. უფრო ეფექტურია დისკის განზრახ დაზიანების მეთოდი, როდესაც დისკის მაგნიტური ზედაპირის გარკვეული უბანი ზიანდება ლაზერის სხივის ან თუნდაც უბრალო ნემსის მეშვეობით. ასეთ შემთხვევაში პრაქტიკულად შეუძლებელია კოპირებული დისკეტის ორიგინალთან იდენტურობის მიღწევა. “გასაღები“ დისკების გამოყენების ძირითადი ნაკლია დისკეტების არასაიმედოობა და მათი დაზიანების დიდი ალბათობა. მეორე ტიპის დაცვაში (ე.ი. ალგორითმის მითვისებისაგან) დიდ როლს თამაშობს საავტორო უფლებებთან დაკავშირებული ვითარება ქვეყანაში. ამ განხრით პროგრამებში ხშირად იყენებენ შიფრირებულ საიდენტიფიკაციო ნიშნულებს, რომლებიც საკმარისი გარანტია საავტორო უფლების დასამტკიცებლად.

ორივე შემთხვევაში დიდ საშიშროებას წარმოადგენენ “ჰაკერები“, რომლებსაც შეუძლიათ პირველ შემთხვევაში ამომიციონ დაცვის ალგორითმი და გააუვნებელყოფონ იგი, ან, მეორე შემთხვევაში, გაარჩიონ თვითონ პროგრამის ალგორითმი რათა მითვისონ იგი. ჰაკერების ძირითადი “იარაღებია“ ე.წ. დებაგერი და დიზასემბლერი, პირველი იძლევა პროგრამის ტრასირების - ნაბიჯ-ნაბიჯ შესრულების შესაძლებლობას, ხოლო მეორე - ასემბლერზე პროგრამის ტექსტის მიღების შესაძლებლობას. ტრასირებისაგან პროგრამის დაცვა შეიძლება, თუკი პროგრამა აკონტროლებს პროგრამულ წყვეტებს INT 1 და INT 3, რომლებსაც გამოიყენებს დებაგერი პირველს პროგრამის ნაბიჯ-ნაბიჯ შესრულებისას ხოლო მეორეს შიფრების ადგილებში (*breakpoints*). დებაგერებისაგან დაცვა ასევე შესაძლებელია კოდის შესრულების დროის ან დაბრუნების მისამართის ანალიზით. ალგორითმის დაცვა დიზასემბლირებისაგან შეიძლება პროგრამის კოდის შიფრაციით, ამ დროს ძირითადი მოდული ჩაიტვირთება და განიშიფრება მეორე, დამხმარე მოდულის მიერ.

ნებისმიერ შემთხვევაში იდეალური დაცვა არ არსებობს, შესაძლებელია ნებისმიერი დაცვის ამოცნობა და უვნებელყოფა. დაცვის სისტემის შეფასებაში ერთადერთ კრიტერიუმად შეიძლება გამოდგეს მისი სირთულე - თუ რა დანახარჯს მოითხოვს დაცვის უვნებელყოფის პროცესი. თუ ეს დანახარჯი მეტია ვიდრე პროგრამის მითვისებით მიღებული სარგებელი, ითვლება რომ დაცვა ეფექტურია.

Методика разработки прикладного программного обеспечения в CAD/CAM

Л. Мегрелишвили

CAD/CAM - это интегрированная система компьютерного проектирования решающая комплекс задач технологического назначения.

На сегодняшний день существует два типа систем CAD/CAM - систем, это системы с закрытой архитектурой и системы с открытой архитектурой. Системы с закрытой архитектурой решают узкий класс задач

ЗАКРЫТАЯ АРХИТЕКТУРА



ПРОГРАММНЫЕ МОДУЛИ
СОЗДАЮТСЯ РАЗРАБОТЧИКОМ

пользователя, системы с открытой архитектурой строятся на базе экспертных систем, что позволяет пользователю самому создавать собственные CAD/CAM- системы.

Разработка CAD/CAM - систем с открытой архитектурой осуществляется в двух укрупненных направлениях:

- Разработка экспертной системы которая реализует метафункции и разрабатывается разработчиком.

ОТКРЫТАЯ АРХИТЕКТУРА



ПРОГРАММНЫЕ МОДУЛИ
СОЗДАЮТСЯ ПОЛЬЗОВАТЕЛЕМ

- Разработка собственной CAD/CAM - системы которая решает конкретные задачи пользователя и разрабатывается пользователем.

Создание собственной CAD/CAM - системы осуществляется посредством создания собственных программных модулей функционирующих в среде экспертной системы.

Целью данного исследования было разработка методики создания программных модулей. В данном докладе рассмотрена методика создания программных модулей для формирования Управляющих Программ (УП) перемещения инструмента в среде экспертной системы **Turbo T**. Целью разработки данной методики является блочно - модульная

структура программирования.

Структурированность определяет правило организации внутри программы связей по передаче данных и управления. Структурированность упрощает и уменьшает объем программы, увеличивает скорость компиляции и интерпретации, повышает эффективность и надежность программирования.

Модуль разделен на отдельные структурные единицы, блоки которые физически реализованы внутри программы, функционируют независимо друг от друга и выполняют часть общей задачи, в частном случае блоку может соответствовать и отдельный программный модуль.

В результате анализа были выделены следующие шесть структурных единиц - блоков:

Первый блок - блок анализа, проверяет и присваивает текущие значения внутренним параметрам.

Второй блок - блок вычисления, вычисляет приращения к опорным точкам для определения опорных точек эквидистанты, число проходов, устанавливает признак конца зоны.

Третий блок - реализует типовые перемещения инструмента.

Четвертый блок - задает перемещение инструмента по эквидистанте и непосредственно по контуру детали.

Пятый блок - блок буфера, запоминает промежуточные результаты.

Шестой блок - блок коррекции режимов обработки.

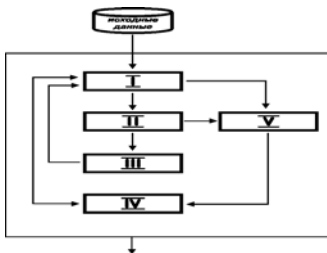
С учетом приведенной выше методике был разработан программный модуль СИКЛ L95 формирующий траекторию перемещения инструмента для многопроходной обработки типовых деталей типа втулка и фланец в среде экспертной системы **Turbo T**.

Входными данными первого блока являются характеристики описывающие контур детали, глубину резания, и припуск под полустовую обработку. Анализируя корректность задания, входные данные

I	Анализа
II	Вычисления
III	Реализации типовых перемещений инструмента
IV	Перемещения по эквидистанте
V	Буфера
VI	Коррекции режима обработки

присваивает их внутренним параметрам.

Функциями второго блока является вычисления приращений к опорным точкам, определение количества проходов и глубины резания, устанавливаются также ограничения по перемещению инструмента и устанавливается признак конца зона, вычисляются также значения центральной точки для окружности, в случае, если перемещение в зоне по окружности. Последним этапом функционирования второго блока является передача определенных значений третьему блоку и помещение некоторых своих значений в блок буфер.



Третий блок реализует перемещение инструмента по схеме петля. Выполнив определенное количество проходов в текущей зоне, передает управление первому блоку который передает характеристики следующей зоны, для обработки второму блоку. Завершив обработку существующего количества зон управление передается первому блоку который устанавливает завершение многопроходной обработки и передает данные и управление четвертому блоку.

Четвертый блок получая значения координат опорных точек от первого блока, приращения для определения опорных точек эквидистанты и значения центральной точки в случае перемещения по окружности, задает перемещение по эквидистанте. После завершения перемещения по эквидистанте, осуществляется перемещение непосредственно по контуру, после завершения перемещения по контуру инструмент возвращается в исходную точку.

Пятый блок хранит результаты вычислений второго блока, приращения к опорным точкам контура и значения центральной точки, и передает их четвертому блоку для осуществления полуступенчатой обработки детали. В режиме компиляции мы получаем инструкции *CLDATA* для многопроходной обработки деталей указанного выше класса.

