Formal Specification المواصفات التحريرية أو الرسمية

هى تقنيات لمواصفات واضحة جلية غير غامضة للبرمجيات. وتنمم مواصفات النظام الرسمية التحريرية تقنيات المواصفات الشفهية غير الرسمية. وهذه المواصفات دقيقة محددة وغير غامضة وبدون التباس، وتستبعد مناطق الشك فى المواصفات، وتجبر على تحليل متطلبات النظام فى مرحلة مبكرة، وتستخدم تقنيات المواصفات الرسمية فى تطوير النظم الحرجة والمعيارية.

تناسب التقنيات الجبرية Algebraic techniques مواصفات واجهة الاستخدام حيث يتم تعريف الواجهة كمجموعة فئات كائن object classes.

تنمذج التقنيات المبنية على النماذج Model-based techniques النظام باستخدام مجموعات ووظائف (دوال) مما يبسط بعض أنواع المواصفات السلوكية Behavioral specification.

تساعد تقنيات المواصفات الرسمية على اكتشاف المشاكل فى متطلبات النظام، وتستخدم التقنيات الجبرية لخصائص الواجهة Interface specification، وتستخدم التقنيات المبنية على النماذج Model-based techniques للمواصفات السلوكية.

## الطرق الرسمية أو التحريرية Formal methods

هى جزء من مجموعة تقنيات عامة تعرف بالطرق الرسمية، وتعتمد كلها على التمثيل الرياضى Mathematical representation وتحليل البرمجيات، وتتضمن الطرق الرسمية: المواصفات الرسمية Formal specification، وتحليل المواصفات والإثبات Specification analysis and proof، والتطوير التحويلى Transformational development، والتحقق من البرنامج Program verification.

لم تصبح الطرق الرسمية الاتجاه السائد فى تقنيات تطوير البرمجيات فور ظهورها، وكانت تقنيات هندسة البرمجيات الأخرى أكثر نجاحا فى زيادة جودة النظام لذلك فإن الحاجة إلى الطرق الرسمية قلت، مع تغيرات السوق أصبحت الاستجابة الفورية للسوق هى العامل الأساسى بدلا من البرمجيات الأقل خطأ وكانت الطرق الرسمية تقلل الوقت للسوق لذلك ازدادت أهميتها، نطاق الطرق الرسمية محدود كما أنها ليست مناسبة تماما لتوصيف وتحليل واجهات المستخدم وتفاعل المستخدم، وأيضا يصعب عليها النهوض بأعباء مع النظم الكبيرة.

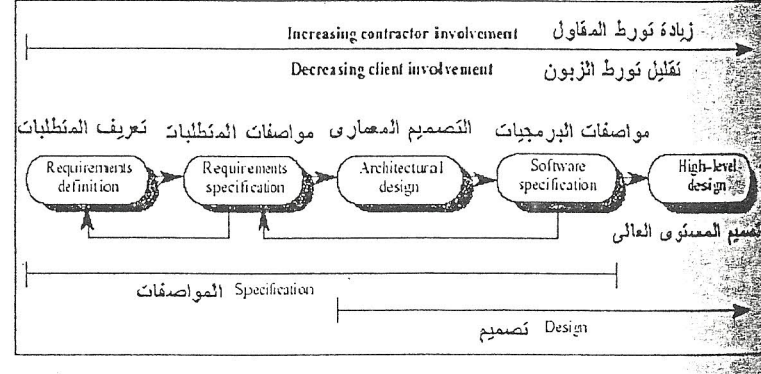
الطرق الرسمية محدودة فى قابلية التطبيق العملية Practical applicability الفائدة الرئيسية لها هى تقليل عدد الأخطاء فى النظام، وبهذا فإن مجال التطبيق الأكثر لها هو فى النظم الحرجة Critical systems، ففى هذه الحالة يصبح استخدام الطرق الرسمية الأكثر قبولا للتكلفة المؤثرة.

## المواصفات فى عمليات البرمجيات

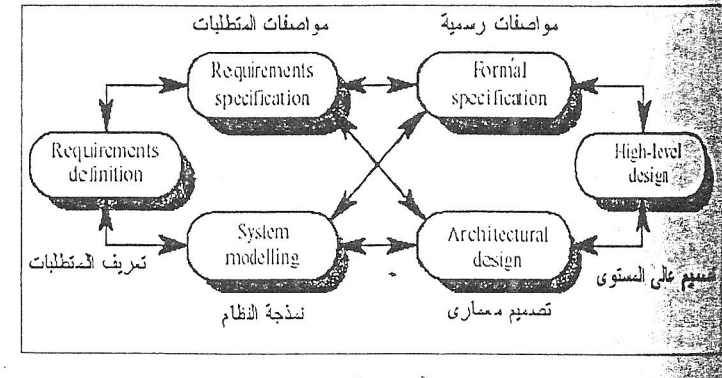
Specification in the software process

تمازج المواصفات Specification مع التصميم Design بطريقة لا يمكن الخلاص منها، ويعد التصميم المعمارى Architectural Design حتميا فى تشييد المواصفات.

يبين الشكل التالى الموصفات والتصميم Specification and Design :



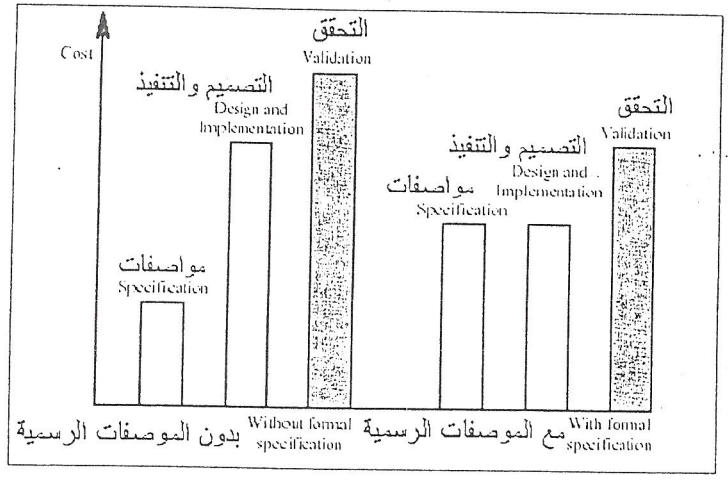

يعبر عن المواصفات الرسمية بتدوين رياضى Mathematical notation مع ألفاظ دقيقة التحديد Precisely Defined Vocabulary، وقواعد Syntax ودلالات Semantics.






## تكاليف التطوير مع مواصفات رسمية

Development costs with formal specification




## تقنيات المواصفات Specification techniques

- المنهج الجبرى Algebraic approach : يتم توصيف النظام بمصطلحات من عملياته والعلاقات بينها.
- المنهجية المعتمدة على النموذج Model-based approach: يتم توصيف النظام بمصطلحات من نموذج حالة يتكون باستخدام إنشاءات رياضية مثل المجموعات والتتابعات Sets and Sequences، وتعرف العمليات بتعديلات فى حالة النظام.

## لغات التوصيف الرسمى Formal specification languages:

| المنهجية | تتابعية SEQUENTIAL | مساعد CONCURRENT |
|---|---|---|
| الجبرية Algebraic | Larch (Guttag, Horning et al., 1985; Guttag, Horning et al., 1993), OBJ (Futatsugi, Goguen et al., 1985) | Lotos (Bolognesi and Brinksma, 1987), |
| معتمدة على نموذج Model-based | Z (Spivey, 1992) VDM (Jones, 1980) B (Wordsworth, 1996) | CSP (Hoare, 1985) Petri Nets (Peterson, 1981) |

## استخدام المواصفات الرسمية

## Use of formal specification

تتضمن المواصفات الرسمية مجهودات كشف وتحر كبيرة فى المراحل الأولى من تطوير البرمجيات، وهذا يقلل من أخطاء المتطلبات لأنها ترغم على التحليل التفصيلى للمتطلبات، ويمكن اكتشاف كلا من عدم الاكتمال أو عدم التماسك وحل مشاكلهما وبالتالى يتم توفير كمية إعادة العمل بسبب تخفيض مشاكل المتطلبات.

## المواصفات الجبرية

تقسم النظم الكبيرة إلى نظم فرعية يتم تطويرها مستقلة عن بعضها البعض، كما تستخدم النظم الفرعية ذاتها نظما فرعية أخرى، وتكون واجهة النظام الفرعى هى الجزء الأساسى فى عملية المواصفات، وما إن يتم تعريف واجهات الاستخدام يمكن تطوير النظم الفرعية بصورة مستقلة.

يتم تعريف واجهات النظم الفرعية كمجموعة من أنواع البيانات الموجزة المجردة أو كائنات. ينجز كل نظام فرعى هذه الواجهات وتصل النظم الفرعية إلى بعضها من خلال هذه الواجهات، ولهذا يكون من المحتم أن تكون واجهة النظام الفرعى واضحة وغير غامضة التوصيف، وهذا يقلل فرص عدم الفهم بين ما توفره النظم الفرعية من إمكانات وبين النظام الفرعى الذى يستخدم هذه الإمكانات.

نقطة البداية فى المواصفات هى المواصفات غير الرسمية للواجهة Informal Interface Specification التى يتم التعبير عنها كمجموعة من أنواع بيانات مجردة abstract data types أو فئات كائن Object classes تم التفاوض بشأنها بين مصممى النظام الفرعى.

تناسب المنهجية الجبرية بصفة خاصة تعريف واجهات النظم الفرعية، وتعرف هذه الطريقة من المواصفات الرسمية فئة الكائن أو نوع البيانات المجردة فى مصطلحات من علاقات بين نوع التشغيلات Type operations.

أول من ناقش هذه المنهجية هو جوتاج 1977 Guttag فى منهجية مواصفات أنواع البيانات المجردة Approach in the specification of abstract data، وأيضا بين كوهين [Cohen, 1986] كيف أن التقنية يمكن أن تتوسع لاستكمال مواصفات النظام باستخدام مثال عن نظام استرداد مستند Document Retrieval system.

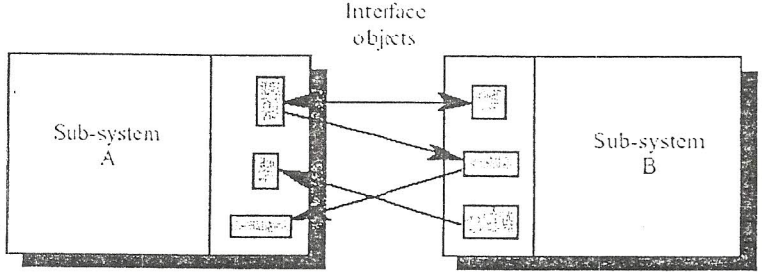


تم تطوير العديد من لغات التوصيف الجبرى بما فيها OBJ [Futatsugi, 1985 and Larch [Guttag, 1985 #163 [#155]، ومن الصعب البرهنة على الاكتمال والتماسك الرياضى للمواصفات الجبرية، ولم تنقص هذه المشاكل من فوائدها فى دعم العمليات الحرجة فى توصيف الواجهات.

## مواصفات الواجهة Interface specification

تقسم النظم الكبيرة إلى نظم فرعية Subsystems بتعريف جيد للواجهات بين هذه النظم الفرعية، يسمح تعريف واجهات النظم الفرعية بالتطوير المستقل للنظم الفرعية المختلفة، وقد تعرف الواجهات كأنواع بيانات موجزة مجردة Abstract data types أو كفئات كائن Object Classes، وتناسب المنهجية الجبرية Algebraic approach للمواصفات الرسمية مواصفات الواجهات بصورة أكثر.

واجهات النظام الفرعى Sub-system interfaces:




## هيكل المواصفات الجبرية Structure of an algebraic specification

كما فى الشكل التالى يتضمن الهيكل شكلا عاما للتوصيف يتكون من:

| < SPECIFICATION NAME > (Generic Parameter)<br>sort < name ><br>imports < LIST OF SPECIFICATION NAMES > |
|---|
| Informal description of the sort and its operations |
| Operation signatures setting out the names and the types of the parameters to the operations defined over the sort |
| Axioms defining the operations over the sort. |

<اسم المواصفات> (معامل عام)

نوع <اسم>

الاستيراد <لائحة أسماء مواصفات>

التوصيف الغير رسمى للنوع وعملياته

توقيعات العملية يضع أسماء وأنواع المعاملات للعمليات المعرفة فى النوع.

البديهيات تعرف العمليات على النوع.

## مكونات المواصفات Specification components

مقدمة Introduction: تعرف نوع (اسم النوع) Sort (type name) وتعلن عن مواصفات أخرى قد تستخدم.

الوصف Description: وصف غير رسمى للعمليات على النوع.

التوقيع Signature: تعريف قواعد العمليات Syntax of operations فى واجهة ومعاملاتها Parameters.

البديهيات أو الحقائق المقررة Axioms: تعريف دلالة العمليات بتعريف البديهيات التى تميز التصرف Characterize behavior.

## المواصفات الجبرية التقليدية

## Systematic algebraic specification

قد تطور المواصفات الجبرية للنظام بطريقة نظامية عن طريق :

- تشييد أو هيكلة المواصفات Specification structuring.
- تسمية المواصفات Specification naming.
- اختيار العملية Operation selection.
- وضع مواصفات العملية الرسمية Informal operation specification.
- تحديد أو تعريف القواعد Syntax definition.
- تعريف البديهية Axiom definition.

عمليات المواصفات Specification operations هى :

عمليات إنشائية Constructor operations بتوصيف العمليات التى تنتج عناصر النوع



عمليات معاينة Inspection operations بتوصيف العمليات التى تقيم عناصر النوع.

لتوصيف سلوك يتم تعريف عمليات المعاينة لكل عملية إنشائية.

### مثال العمليات على قائمة Operations on a list ADT:

فى هذا المثال تعريف لائحة تضاف العناصر إلى نهايتها وتحذف من المقدمة عن طريق إنشاء Create يجلب قائمة فارغة للوجود، وعملية إنشاء قائمة جديدة Cons برقم مضاف، ويحسب الطول Length حجم القائمة، ويحسب الرأس Head الذى عنصر المقدمة فى القائمة، ويقوم الذيل Tail بإنشاء قائمة باستبعاد عنصر التصدير من القائمة.

العمليات الإنشائية التى تقيم فرز قائمة هى :

- عمليات الإنشاء Create , Cons and Tail.
- عمليات المعاينة التى تأخذ قائمة الفرز كمعامل وتعيد بعد الفرز الطول والتصدير Head and Length.

يمكن تعريف النهاية Tail باستخدام تركيب مبسط للإنشاء Create and Cons، وليست هناك حاجة لتعريف التصدير والطول مع الذيل.

### مواصفات مثال القائمة List specification:

يبين الشكل التالى مواصفات مثال القائمة :

LIST ( Elem )

**sort** List
**imports** INTEGER

Defines a list where elements are added at the end and removed from the front. The operations are Create, which brings an empty list into existence, Cons, which creates a new list with an added member, Length, which evaluates the list size, Head, which evaluates the front element of the list, and Tail, which creates a list by removing the head from its input list. Undefined represents an undefined value of type Elem.

Create → List
Cons (List, Elem) → List
Head (List) → Elem
Length (List) → Integer
Tail (List) → List

Head (Create) = Undefined **exception** (empty list)
Head (Cons (L, v)) = If L = Create **then** v **else** Head (L)
Length (Create) = 0
Length (Cons (L, v)) = Length (L) + 1
Tail (Create ) = Create
Tail (Cons (L, v)) = If L = Create **then** Create **else** Cons (Tail (L), v)