Просмотрели еще раз патент 5995958. Вообще у него интересный способ построения: а именно по сути преобразования патентования парадигмы в патентование аппарата (механизма). Там просто в первом и единственном независимом пункте написано следующее:

Компьютерный аппарат имеющий средства работы с памятью (memory means), взаимодействия (interface means) с другими аппаратами (сюда как я понимаю и пользователи включаются), и реализации (implementation means), включающий базу данных хранящейся в этой памяти. База включает : и идет описание парадигмы (модели).

Соответственно все memory means, interface means и implementation means идут в зависимых пунктах (то есть по идее их вообще может не быть). В принципе таким способом можно по идее запатентовать любую модель программирования (задания информационных систем, БД), и соответственно также наверное может иметь смысл строить наш патент. Другое дело, что патент 2000 года и может тогда были другие правила и сейчас такое не прокатит.

Сразу оговоримся, что claim'ы будут очень подробные, для понимания общей структуры технологии. Впоследствии их придется сливать, возможно даже первые 20 пунктов, что впрочем не важно потому как без одного из первых 20 у конкурирующей системы могут быть существенные проблемы на рынке.

Заявляется :

1. Компьютерная аппарат имеющий средства работы с памятью (memory means), включающий базу данных (вот здесь может имело бы смысл использовать более общий термин, потому как *база данных* предполагает постоянное хранение, то есть память не может быть оперативной, но такого термина пока придумать не можем), хранящуюся в said памяти, интерфейс взаимодействия с пользователем и внешними аппаратами, и средства реализации (implementation means – как реализуется эта система, то есть как задается модель, и какие алгоритмы внутри).

Said база данных состоит из множества объектов и свойств; каждое из said свойств ставит в соответствие нескольких объектам один объект (в общем то это определение функции, но она предполагает вычисление, свойство же более общее понятие и подразумевает в том числе хранение в базе как первичные данные, см. пункте а). с другой стороны для функции уже есть терминология параметров и вычислений, так что вопрос что именно использовать), и создается при помощи функционалов: (надо ли в claim'е их описывать, тем более подробно чего точно не сделаешь, или достаточно их перечислить?)

* 1. External – значения свойства хранятся в базе данных, и изменяются в результате взаимодействия с пользователями и внешними аппаратами.
  2. Composition – подставляет значения одних свойств на вход другим.
  3. Grouping – итерирует все объекты в системе, вычисляя сумму или максимум одного свойства в разрезе других.
  4. Recursion – подставляет одно свойство на вход другому заданное количество раз.

1. (undef’ы) Компьютерный аппарат в claim 1, в котором свойство может иметь значение - не определено (undef). При подстановке undef на вход свойству значение его также должно быть undef. (*инвариант определенности*  - непонятно включать или нет в claim и в патент вообще).
2. (типы объектов) Компьютерный аппарат в claim 1, в котором объект может быть одного из следующих типов: пользовательский объект, целое число, плавающее число, дата, boolean (нету русского аналога). (вопрос надо ли их тут описывать)
3. (арифметика) Компьютерный аппарат в claim 3, в котором множество функционалов для создания свойств включает в себя:
   1. Arithmetic – выполняет арифметические операция (сложения, умножения и т.п.)
4. (сравнение) Компьютерный аппарат в claim 2 и 3, в котором множество функционалов для создания свойств включает в себя:
   1. Compare – если объекты удовлетворяют условию сравнения, возвращает объект Истина типа Boolean, иначе undef.
5. (работа с undef’ами) Компьютерный аппарат в claim 2, в котором множество функционалов для создания свойств, включает в себя:
   1. Union – если все объекты undef - то значение undef, иначе undef'ы игнорирует и суммирует, максимизирует или выбирает последнее не undef значения.
   2. And – в зависимости от определенности остальных переданных объектов, возвращает или объект или значение undef.
6. (ограничения) Компьютерный аппарат в claim 2, который ограничивает ввод свойств созданных при помощи функционала External в соответствии с ограничениями, включающими:
   1. определение свойств в базе данных, для которых значение всегда должно быть undef.
7. (классы и наследование) Компьютерный аппарат в claim 2 и 3, в котором для каждого объекта пользовательского типа, должен задаваться класс.
8. (задание классов в External) Компьютерный аппарат в claim 8, в котором при использовании функционала External для значения и каждого параметра, необходимо задавать класс.
9. (ограничения по классам) Компьютерный аппарат в claim 7 и 9, который ограничивает ввод свойств созданных при помощи функционала External в соответствии с ограничениями, включающими:
   1. принадлежность значения и каждого параметра соответствующим классам.
10. (наследование) Компьютерный аппарат в claim 10, в котором классы можно наследовать друг от друга. Соответственно если класс A наследуется от класса B, то на все объекты класса A действуют все те же ограничения которые действуют на объекты класса B (вообще классическое определение наследования конечно другое, здесь определение в контексте предлагаемой модели)
11. (классифицирующий функционал) Компьютерный аппарат в claim 11, в котором множество функционалов для создания свойств, включает в себя:
    1. Classify – в случае принадлежности параметров заданным классам возвращает заданный объект, и undef – в обратном случае
12. (абстрактные классы) Компьютерный аппарат в claim 11, в котором можно создавать и использовать абстрактные классы – классы, объектов которых не может быть. (соответственно нужны только для использования в claim 9 и далее в интерфейсах, то есть правильнее может что классы делятся на конкретные и абстрактные и в claim 8 идет уточнение - *задавать* ***конкретный*** *класс*, но я не знаю как это оформить)

(теоретически самый базис предлагаемой технологии, дальше уже более конкретные уточнения, впрочем без них применять на практике эту технологию практически не возможно, так как они используют особенности SQL и типовые задачи)

1. (максимизация по нескольким параметрам) Компьютерный аппарат в claim 1, в котором множество функционалов для создания свойств, включает в себя:
   1. Multi-Grouping – вычисляет максимум свойства в разрезе других свойств, при условии, что одно или более дополнительных свойств имеют максимальные значения для тех же объектов. (очень важно при построении списков транзакций)
2. (вычисление по порядку – накапливающаяся сумма, предыдущая) Компьютерный аппарат в claim 1, в котором множество функционалов для создания свойств, включает в себя:
   1. Ordering – вычисляет в разрезе других свойств сумму или последнее значение для всех объектов, значение заданных свойств которых, предшествует значению этих же свойств для переданных объектов. (чуть более сложный функционал не так просто без картинки и формализмов описать. Имеет что-то общее с analytic functions в SQL)
3. (механизмы ввода не External свойств) Компьютерный аппарат в claim 1, включающий интерфейсы ввода не External свойств, и их реализацию в зависимости от используемого функционала. (слишком абстрактный пункт возможно)
4. (механизмы ввода суммирующего группирующего свойства) Компьютерный аппарат в claim 16, в котором при использовании Grouping функционала для вычисления суммы, можно задавать ограничение сверху и порядок, в соответствие с которыми определяется реализация интерфейса ввода создаваемого свойства (в claim 16, как именно в claim'е наверное нет смысла приводить, это будет внутри патента).
5. (для реализации обратных связей : one-to-one) Компьютерный аппарат в claim 7, в котором при использовании Grouping функционала, можно ограничить что в группируемой выборке, должен быть ровно один набор объектов. (тут надо наверное построить предложение как в claim'ах 7 и 10, то есть когда идет ввод External свойств и т.п., но что-то не придумали как построить предложение)
6. (для ввода one-to-one связей) Компьютерные аппарат в claim 16 и 18, в котором для Grouping функционалов с одним набором объектов, можно задавать вводимое свойство, в соответствие с которыми определяется реализация интерфейса ввода создаваемого свойства (в claim 16, как именно в claim'е наверное нет смысла приводить, это будет внутри патента)
7. Компьютерный аппарат в claim 16, в котором для свойств, созданных при помощи Union функционалов на операцию последнее определенное, интерфейс ввода реализуется как первый из интерфейсов, который в состоянии обработать, требуемое изменение.

Еще раз повторюсь, что излишняя конкретность последних 7 пунктов – это иллюзия, без них конечная система будет не жизнеспособна, поэтому если это нам поможет существенно сократить время патентования (за счет, например доказательства изобретательского уровня) то возможно будет иметь смысл включить побольше пунктов в независимые.

Собственно это начальные 20 пунктов, без механизмов производных изменений, custom действий, интерфейсов взаимодействия с пользователем и всего что связано с реализацией (как именно все работает) – постоянно хранимые свойства, задание физической модели, алгоритмы, впрочем последнее может не иметь смысл патентовать, если самой модели будет достаточно. Все последние механизмы будут более конкретными, хотя впрочем и не менее важными. Но прежде чем продолжать хотелось бы согласовать насколько все возможно \ имеет смысл патентовать в таком виде как мы сейчас описали, может имеет смысл кардинально изменить подход, в каком-то другом направлении, о котором мы не знаем. Так что хотелось бы в ближайшее время услышать оценку с Вашей стороны, не является ли наш первый блин формулы комом.