В своем выступлении Бретт Виктор рассматривает эволюцию программирования и его развитие в течение следующих сорока лет. Автор подмечает, что встреча с компьютерами происходит во всех сферах человеческой жизни – от фундаментальных научных открытий до финансового сектора и управления деньгами. Технологии меняются очень быстро, однако мышление людей развивается с меньшей скоростью. Это, в свою очередь, создает бОльшую проблему – сопротивление идеям и отдача предпочтения старому, привычному, несмотря на объективное улучшение. Как люди не доверяли и не хотели пользоваться поездами, так и для программистов потребовалось большое количество времени для перехода от работы на машинном коде к ассемблеру и далее к языкам высокого уровня. Брет Виктор, опираясь на исследования ХХ века, выделяет четыре ключевых направления, которые могут изменить подход к программированию.

Первой высказанной идеей является прямое манипулирование данными. Спикер предвидит будущее, где через визуализационный (описательный) подход пользователи смогут взаимодействовать с данными, позволяя системе неявно генерировать код. Например, в системе Sketchpad Айвена Сазерленда компьютер автоматически корректировал чертеж в зависимости от запросов пользователя – это интуитивный диалог с машиной, который значительно упрощает и ускоряет процесс.

Отдельное внимание было уделено программированию с использованием целей. Виктор подчеркнул значимость перехода к высокоуровневым задачам и оставлению написания подробных процедур, как это реализовано в Planner и Prolog. В этих системах компьютер сам ищет способ достижение цели - вместо императивного описания действий, используется декларативный подход с использованием сопоставления с образцом или логическим выводом. Данный подход приведет к макроуправлению и «общению» программ без определенных интерфейсов.

Далее обсуждалось пространственное представление информации, которое коррелирует с первой идеей. Виктор считает, что в будущем произойдет переход к визуальному и пространственному программированию, уходя от текстового, что исторически связано с ограничениями старых устройств ввода-вывода. Такие системы, как NLS, GRAIL и Smalltalk позволяют визуализировать иерархию, графы зависимостей и интерактивные диаграммы, что является более интуитивным и наглядным, а это, в свою очередь, улучшает понимание и взаимодействие с кодом.

Под критику также было поставлено и последовательная модель программирования, которая привязана к архитектуре фон Неймана (один процессор + большая память), предсказывая переход к параллельному оборудованию. Старый подход ограничивает мышление программистов в последовательные рамки, но физика микросхем требует другой архитектуры, где тысячи транзисторов могут работать одновременно. В частности, модель Акторов представляет асинхронный обмен сообщениями.

Бретт Виктор особенно подчеркивает необходимость открытости к новым идеям и не зацикливания на устаревших парадигмах. История с нежеланием программистов IBM 650 переходить с машинного кода на ассемблер является ярким примером. Функциональное программирование, ООП и другие парадигмы – это не окончательная истина. Только открытость к экспериментам позволит развиваться любой науке – это залог прорыва. Лишь после нашего признания того, что мы не знаем, что такое программирование как таковое, мы сможем по-настоящему творить и создавать будущее, переходя к новым мощностям, новой доступности и новой естественности.