

谈谈低代码与AI大模型

在前几年的工作中，从零开始搞了一个低代码平台，但是由于之前公司投入的资源一直比较少，进度和效果一直不如人意。最近离职后，终于有时间来做一些自己想做的事情，打算把前几年的低代码这块的积累好好梳理一下，手撸一个低代码平台，作为自己以后业余的兴趣爱好，同时也计划逐步将平台的源代码开源到GitHub，希望希望有更多的人来一起完善这个平台。

近几年AI大模型大火，很多人都在说现在大模型已经很强大了，整个系统都能给你直接生成，现在搞低代码还有前途吗？

因此在开始谈低代码之前，我觉得有必要先探讨一下低代码与AI大模型之间的关系。

1. AI大模型的不足

目前的AI大模型的核心其实是AIGC，其严格意义上来说，并不是真正意义的AI，而是通过全世界海量的知识训练，使用其概率模型来预测并生成相关的输出，因为其知识面足够大足够广，所以在很多通用领域，其生成的内容的准确率非常高。但是也有不少实际的应用场景中，AI大模型存在很多的不足之处。

首先，因为其核心本质是一种概率模型，其生成内容的准确性以及稳定性并不能得到保障。一些对准确性和稳定性要求较高的场景，目前大模型的应用并不理想。另外对于一些需要有一定的创造性能力的场景，大模型也并不是太能满足。

其次，AI大模型的上下文大小受限，面对一些复杂的场景，比如实际的软件系统工程，参与的人员非常多，流程也非常复杂，如果完全依靠大模型的话，体系性支撑的能力比较有限。目前大模型在软件开发领域主要还是辅助开发为主，比如一些代码插件。

再次，不知道从什么时候开始，对话的模式成了AI大模型的标配，给人的感觉就是只有用了对话的模式才算用了大模型。其实个人感觉对大部分应用场景来说，对话模式是一种很低效的交互模式。

最后，训练大模型的知识以通用知识为主，各个垂直领域的知识相对有限，虽然通过微调和RAG能够解决一定的问题，但是相对来说，垂直领域应用场景的支撑能力还是要差一点。

2. 低代码与AI大模型能够形成互补关系

回过头来，咱们再来看看低代码平台的本质是什么？个人认为，低代码通过对常用的技术组件、代码逻辑、功能组件以及开发场景的抽象，底层采用组件封装、批处理生成，前端通过界面化、拖拉拽等方式/手段，降低使用人员在开发语言、技术、框架等方面的学习成本以及使用成本，并提高开发效率的开发工具。

软件开发领域本身就是一个专业性比较强的领域，开发的目标系统往往又是业务知识比较强的垂直领域，而且任何一个具备实用性的实际业务系统，相对目前的大模型（除非大模型有质的突破或者能够成为真正的AI）来说，其复杂度都是远远超出其上下文能够承载的程度的，因此单靠AI大模型本身是很难进行系统性解决的。而通过低代码平台以及使用低代码平台的人的介入，可以很好的解决AI大模型的不足，因此个人认为，在软件开发领域，AI大模型与低代码并不是直接竞争或者替代的关系，反而通过两者的结合能够形成相辅相成的互补关系。

（1）低代码平台能够提供系统性的组织和管理能力，为AI大模型提供上下文不足以及体系性管理能力缺乏的不足。

（2）在低代码平台中集成AI大模型的使用能力，相比目前的智能体开发平台来说，能够提供更深入应用大模型的能力。

（3）低代码平台便捷的界面化、拖拉拽能力，能够有效的弥补AI大模型低效的对话交互模式，尤其是在初始生成之后的调整场景。

（4）在低代码平台中深入结合AI大模型强大的意图识别能力，可以使低代码平台生成功能/业务逻辑的完成度更高、更智能。

3. 为什么需要低代码

虽然低代码早就已经过了暴火的时候，市面上也已经有很多已经很成熟的低代码平台，但到目前为止，各大软件厂商对低代码的应用还停留在喊口号、试用、开发demo或简单功能开发为主的阶段，实际的应用很不理想。

但是不管实际的应用效果如何，大环境的现实已经非常严酷的摆在那儿，经济大环境形势严峻、失业、消费降级、甲方预算一降再降、人力成本居高不下，除了裁员这种杀敌一千，自损八百的暴力手段之外，如何降低开发成本、提高开发效率已经是各软件企业不得不考虑的一个事情。而不管是低代码，还是AI大模型都是不错的选择方向，那么AI+低代码的结合，会不会产生更好的效果？

4. 设计低代码平台的关键

下面简单的列几个设计和实现低代码平台必须要考虑的几个关键点（个人观点）：

（1）低代码平台不是万能的。必须要承认低代码平台解决不了所有的问题，在设计的时候就需要考虑与已有系统/历史代码的结合问题。

（2）低代码平台不能只解决简单问题。低代码平台不能只解决Demo/简单功能的开发，必须能解决一些实际的复杂场景。

(3) 低代码平台需要有较合理/先进的设计和对象组织管理理念，介于对整个复杂的系统业务进行有效的管理。

(4) 具备持续迭代升级/扩展能力，包括低代码本身和基于低代码平台开发的系统。

(5) 拥抱AI浪潮，一方面使AI的能力能够更无缝的结合到业务系统的方方面面，另一方面也可以使低代码平台本身能够更加的智能。

5. 其他

后续将通过更多的文章，逐步对低代码平台的设计理念、思想以及具体的设计进行细化，随着设计与开发的进展，低代码平台的源码将开源到GitHub：<https://github.com/zjyzju/zPaaS-lowcode>，敬请关注。

如果有低代码平台以及AI相关的讨论，可以发送邮件到我的邮箱：zjyzju@163.com。